National Security Agency
Information Assurance
Directorate

(U) Global Information Grid Information Assurance Capability/Technology Roadmap

Version 1.0 (Final Draft)

26 October 2004

Prepared by:
IA Architecture Office (I11)

(U) This page intentionally left blank

(U) TABLE OF CONTENTS

Section	Title	Page
	(U) Revision History	**xv**
	(U) Executive Summary	**I**
1	**(U) Introduction**	**1-1**
1.1	**(U) Purpose**	**1-1**
1.2	**(U) Scope**	**1-2**
1.3	**(U) Approach**	**1-2**
2	**(U) IA System Enablers and their Technologies**	**2-1**
2.1	**(U) Identification and Authentication**	**2.1-1**
2.1.1	(U) GIG Benefits due to I&A	2.1-2
2.1.2	(U) I&A: Description	2.1-2
2.1.3	(U) I & A: Technologies	2.1-4
2.1.3.1	(U) Authentication Tokens	2.1-5
2.1.3.2	(U) Biometrics	2.1-15
2.1.3.3	(U) Device/Service Authentication	2.1-25
2.1.3.4	(U) Authentication Protocols	2.1-35
2.1.3.5	(U) Authentication Confidence	2.1-43
2.1.3.6	(U) Single Sign-On	2.1-46
2.1.4	(U) I&A Gap Analysis	2.1-59
2.1.5	(U) Identification and Authentication: Recommendations and Timelines	2.1-62
2.2	**(U) Policy-Based Access Control**	**2.2-1**
2.2.1	(U) GIG Benefits due to Policy-Based Access Control	2.2-2
2.2.2	(U) Policy-Based Access Control: Description	2.2-2
2.2.2.1	(U) Core RAdAC Functions	2.2-2
2.2.2.2	(U) Assured Metadata and Data Describing Enterprise Elements	2.2-4
2.2.2.3	(U) Digital Access Control Policy	2.2-5

2.2.2.4 (U) IA Enabler Dependencies 2.2-6
2.2.3 (U) Policy-Based Access Control: Technologies 2.2-6
2.2.3.1 (U) Core RAdAC........ 2.2-7
2.2.3.2 (U) Assured Metadata 2.2-15
2.2.3.3 (U) Digital Access Control Policy........ 2.2-40
2.2.4 (U) Distributed Policy Based Access Control: Gap Analysis........ 2.2-44
2.2.4.1 (U) Core RAdAC: Gap Analysis........ 2.2-44
2.2.4.2 (U) Assured Metadata: Gap Analysis........ 2.2-46
2.2.4.3 (U) Digital Access Control Policy: Gap Analysis........ 2.2-49
2.2.5 (U) Policy Based Access Control: Recommendations and Timelines...... 2.2-50
2.3 (U) Protection of User Information 2.3-1
2.3.1 (U) GIG Benefits Due to Protection of User Information 2.3-2
2.3.2 (U) Protection of User Information: Description........ 2.3-3
2.3.3 (U) Protection of User Information: Technologies........ 2.3-7
2.3.3.1 (U) Technologies for Protecting Data-at-Rest........ 2.3-8
2.3.3.2 (U) Technologies for Protecting Data-in-Transit 2.3-10
2.3.3.3 (U) Trusted Platforms........ 2.3-87
2.3.3.4 (U) Trusted Applications........ 2.3-98
2.3.3.5 (U) Cross Domain Solution Technologies 2.3-106
2.3.3.6 (U) Non-Repudiation........ 2.3-116
2.3.4 (U) Protection of User Information: Gap Analysis........ 2.3-126
2.3.5 (U) Protection of User Information: Recommendations and Technology Timelines........ 2.3-130
2.3.5.1 (U) Data-in-Transit........ 2.3-130
2.3.5.2 (U) Cross Domain Solutions 2.3-132
2.4 (U) Dynamic Policy Management........ 2.4-1
2.4.1 (U) GIG Benefits due to Dynamic Policy Management........ 2.4-2

2.4.2 (U) Dynamic Policy Management: Description 2.4-2
2.4.3 (U) Dynamic Policy Management: Technologies 2.4-6
2.4.3.1 (U//FOUO) Development of Policies 2.4-6
2.4.3.2 (U) Distribution of Policies 2.4-22
2.4.3.3 (U) Policy Management Architectures 2.4-29
2.4.4 (U) Dynamic Policy Management: Gap Analysis 2.4-31
2.4.5 (U) Dynamic Policy Management: Recommendations and Timelines 2.4-33
2.4.5.1 (U) Standards 2.4-33
2.4.5.2 (U) Technology 2.4-34
2.4.5.3 (U) Infrastructure 2.4-34
2.5 (U) Assured Resource Allocation 2.5-1
2.5.1 (U) GIG Benefits of Assured Resource Allocation 2.5-3
2.5.2 (U) Assured Resource Allocation: Description 2.5-3
2.5.3 (U) Technologies 2.5-5
2.5.3.1 (U//FOUO) IA Policy-Based Routing 2.5-6
2.5.3.2 (U//FOUO) Operational-Based Resource Allocation 2.5-17
2.5.3.3 (U//FOUO) Integrity of Network Fault Monitoring/Recovery and Integrity of Network Management & Control 2.5-26
2.5.4 (U) Assured Resource Allocation: Gap Analysis 2.5-38
2.5.5 (U) Assured Resource Allocation: Recommendations and Technology Timelines 2.5-40
2.6 (U) Network Defense and Situational Awareness 2.6-1
2.6.1 (U) GIG Benefits due to Network Defense and Situational Awareness 2.6-2
2.6.2 (U) Network Defense and Situational Awareness: Description 2.6-3
2.6.3 (U) Network Defense and Situational Awareness: Technologies 2.6-8
2.6.3.1 (U) Protect Technologies 2.6-9
2.6.3.2 (U) Deception Technologies 2.6-14

2.6.3.3 (U) Situational Awareness 2.6-21
2.6.3.4 (U) Network Mapping 2.6-26
2.6.3.5 (U) Intrusion Detection Systems 2.6-30
2.6.3.6 (U) Intrusion Prevention Systems (IPSs) 2.6-37
2.6.3.7 (U) User Activity Profiling 2.6-39
2.6.3.8 (U) Cyber Attack Attribution 2.6-44
2.6.3.9 (U) Correlation Technologies 2.6-49
2.6.3.10 (U) CND Response Actions 2.6-54
2.6.3.11 (U) Automated IAVA Patch Management 2.6-58
2.6.4 (U) Network Defense and Situational Awareness: Gap Analysis 2.6-63
2.6.5 (U) Network Defense and Situational Awareness: Recommendations and Timelines 2.6-66
2.6.5.1 (U) Standards 2.6-66
2.6.5.2 (U) Technology 2.6-66
2.6.5.3 (U) Infrastructure 2.6-69
2.6.5.4 (U) Technology Timelines 2.6-69
2.7 (U) Management of IA Mechanisms and Assets 2.7-1
2.7.1 (U) GIG Benefits due to Management of IA Mechanisms and Assets 2.7-1
2.7.2 (U) Management of IA Mechanisms and Assets: Description 2.7-1
2.7.2.1 (U) Identity Management 2.7-2
2.7.2.2 (U) Privilege Management 2.7-5
2.7.2.3 (U) Key Management 2.7-9
2.7.2.4 (U) Certificate Management 2.7-11
2.7.2.5 (U) Configuration Management of IA Devices and Software 2.7-14
2.7.2.6 (U) Inventory Management 2.7-16
2.7.2.7 (U) Compromise Management of IA Devices 2.7-16
2.7.2.8 (U) Audit Management 2.7-17

2.7.3 (U) Management of IA Mechanisms & Assets: Technologies 2.7-18
2.7.3.1 (U) Identity Management 2.7-18
2.7.3.2 (U) Privilege Management 2.7-26
2.7.3.3 (U) Key Management 2.7-33
2.7.3.4 (U) Certificate Management 2.7-49
2.7.3.5 (U) Configuration Management of IA Devices and Software 2.7-59
2.7.3.6 (U) Inventory Management 2.7-68
2.7.3.7 (U) Compromise Management of IA Devices 2.7-76
2.7.3.8 (U) Audit Management 2.7-82
2.7.4 (U) Management of IA Mechanisms & Assets: Gap Analysis 2.7-93
2.7.4.1 (U) Identity Management 2.7-96
2.7.4.2 (U) Privilege Management 2.7-96
2.7.4.3 (U) Key Management 2.7-97
2.7.4.4 (U) Certificate Management 2.7-97
2.7.4.5 (U) Configuration Management of IA Devices and Software 2.7-98
2.7.4.6 (U) Inventory Management 2.7-99
2.7.4.7 (U) Compromise Management of IA Devices 2.7-99
2.7.4.8 (U) Audit Management 2.7-99
2.7.5 (U) Management of IA Mechanisms and Assets: Recommendations and Timelines 2.7-100
2.7.5.1 (U) Standards 2.7-100
2.7.5.2 (U) Technology 2.7-101
2.7.5.3 (U) Infrastructure 2.7-101
3 (U) Summary 3-1
3.1 (U//FOUO) Assured information Sharing Summary 3.1-3
3.1.1 (U) Identification and Authentication Technologies 3.1-3
3.1.2 (U) Access Control and Data Labeling Technologies 3.1-5

3.1.3 (U) Cross-Domain Technologies 3.1-7
3.1.4 (U) Trusted Platform Technologies 3.1-9
3.2 (U) Highly Available Enterprise Summary 3.2-11
3.2.1 (U//FOUO) IA Policy-based Routing for Mobile/Tactical Environments Technologies 3.2-11
3.2.2 (U) End-to-End Resource Allocation Technologies 3.2-12
3.2.3 (U//FOUO) Edge-to-Edge Boundary Protection Technologies 3.2-13
3.2.4 (U) Secure Voice Technologies 3.2-13
3.2.5 (U) Enforcement of QoP in Transit Technologies 3.2-14
3.2.6 (U//FOUO) Protection of High Risk Link Technologies 3.2-14
3.3 (U) Assured Enterprise Management and Control Summary 3.3-15
3.3.1 (U) Identity Management Technologies 3.3-16
3.3.2 (U) Inventory Management Technologies 3.3-16
3.3.3 (U) Privilege Management Technologies 3.3-17
3.3.4 (U) Key Management Technologies 3.3-18
3.3.5 (U) Certificate Management Technologies 3.3-19
3.3.6 (U) Configuration Management Technologies 3.3-20
3.3.7 (U) Policy Management Technologies 3.3-20
3.3.8 (U) Audit Management Technologies 3.3-22
3.3.9 (U) Confidentiality & Integrity of Network Management & Control Technologies 3.3-23
3.4 (U) Cyber Situational Awareness and Network Defense Summary 3.4-24
3.4.1 (U) Protection Technologies 3.4-25
3.4.2 (U) Monitoring Technologies 3.4-25
3.4.3 (U) Detection Technologies 3.4-26
3.4.4 (U) Analysis Technologies 3.4-28
3.4.5 (U) Response Technologies 3.4-29

4 (U) Acronyms and Abbreviations 4-1

Appendices

(U//FOUO) Appendix A: Mapping of technologies to IA System Enablers A-2
(U//FOUO) Appendix B: TV-1 for IA A-6
(U//FOUO) Appendix C: TV-2 for IA A-23

(U) LIST OF FIGURES

Figure	Title	Page
Figure 1.3-1:	(U) GIG Mission Concepts, IA Cornerstones, and IA System Enablers	1-3
Figure 1.3-2:	(U) Iterative Development of the GIG IA Capability/Technology Roadmap	1-5
Figure 2.1-1:	(U) Examples of time-driven hardware tokens	2.1-6
Figure 2.1-2:	(U) DoD Common Access Card	2.1-10
Figure 2.1-3:	(U) Example of a Hybrid Device	2.1-14
Figure 2.1-4:	(U) Biometric System Block Diagram	2.1-15
Figure 2.1-5:	(U) Network Authentication Framework	2.1-37
Figure 2.1-6:	(U) Device Authentication Framework	2.1-37
Figure 2.1-7:	(U) Centralized Architecture for Single Sign-On	2.1-48
Figure 2.1-8:	(U) Federated KEBEROS Based Single Sign-On	2.1-50
Figure 2.1-9:	(U) Federated PKI-based Single Sign-on	2.1-51
Figure 2.1-10:	(U) Federated SAML-Based Single Sign-On	2.1-52
Figure 2.2-1:	(U) RAdAC Functional Model	2.2-3
Figure 2.2-2:	(U) Codifying the Net-Centric Data Strategy	2.2-22
Figure 2.2-3:	(U) Encapsulation Notional Diagram	2.2-38
Figure 2.2-4:	(U) Policy-Based Access Control Gap Closure Timelines	2.2-51
Figure 2.3-1:	(U) Context of Non Real-Time Application Security	2.3-11
Figure 2.3-2:	(U) Layered Protocol Wrapping Concept	2.3-12
Figure 2.3-3:	(U) CMS Supports S/MIME and Other Secure Applications	2.3-16
Figure 2.3-4:	(U) TLS Handshake Protocol	2.3-23
Figure 2.3-5:	(U) Model for Web Services Security	2.3-30
Figure 2.3-6:	(U) FNBDT Location in Network Protocol Stack	2.3-33
Figure 2.3-7:	(U) Packet Jitter Mitigation Process	2.3-35
Figure 2.3-8:	(U//FOUO) FNBDT Frame Structure for Signaling Reliability and Reliable	

Transport Data Mode 2.3-36

Figure 2.3-9: (U//FOUO) FNBDT 2400 bps MELP Blank and Burst Superframe Structure 2.3-37

Figure 2.3-10: (U//FOUO) FNBDT 7200 bps G.729D Superframe Structure 2.3-37

Figure 2.3-11: (U) Media Gateway Protocol Stack Illustration 2.3-41

Figure 2.3-12: (U//FOUO) Secure Voice Gateway Functionality 2.3-42

Figure 2.3-13: (U) Real-Time Protocol 2.3-46

Figure 2.3-14: (U) RTCP Sender Report Format- Sender Report (SR) 2.3-48

Figure 2.3-15: (U) SRTP Format 2.3-50

Figure 2.3-16: (U) SRTCP Format 2.3-50

Figure 2.3-17: (U//FOUO) FNBDT over V.150.1 Modem Relay 2.3-52

Figure 2.3-18: (U) V.150.1 Simple Packet Relay Transport for IP networks 2.3-52

Figure 2.3-19: (U//FOUO) State Variable Stepping 2.3-62

Figure 2.3-20: (U) SIP Architecture 2.3-78

Figure 2.3-21: (U) H.323 Network Elements 2.3-80

Figure 2.3-22 (U) Legacy Manifestation of Cross-Domain Solutions 2.3-106

Figure 2.3-23: (U) Controlled Interface Example 2.3-107

Figure 2.3-24: (U) Two MSL Architectures 2.3-108

Figure 2.3-25: (U) Technology Timeline for Protection of User Information: Date in Transit 2.3-132

Figure 2.3-26: (U) Technology Timeline for Protection of User Information: Cross Domain Solutions 2.3-133

Figure 2.4-1: (U) Notional Architectural Framework for Dynamic Policy Management 2.4-3

Figure 2.4-2: (U) Berners-Lee’s Seven Layer Model of the Semantic Web 2.4-19

Figure 2.4-3: (U) Technology Timeline for Dynamic Policy Management 2.4-35

Figure 2.5-1: (U//FOUO) The Role and Components of Assured Resource Allocation 2.5-2

Figure 2.5-2: (U//FOUO) IA Policy-Based Routing 2.5-6

Figure 2.5-3: (U//FOUO) Security-Aware ad-hoc Routing (SAR) in Tactical Wireless Application 2.5-10
Figure 2.5-4: (U) OSPF Implemented With (QoS) IA Policy-Based Routing Extensions........ 2.5-14
Figure 2.5-5: (U) DeSiDeRaTa Architecture for Operational-Based Resource Allocation 2.5-18
Figure 2.5-6: (U) Joint Resource Allocation Across GIG Networks........ 2.5-21
Figure 2.5-7: (U) Basic Elements of SNMP Operation 2.5-26
Figure 2.5-8: (U) SNMPv3 Security Capabilities........ 2.5-27
Figure 2.5-9: (U) SNMPv3 Message Format & Security Components........ 2.5-28
Figure 2.5-10: (U) SNMPv3 View-based Access Control Model (VACM) Logic 2.5-29
Figure 2.5-11: (U) Technology Timeline for Assured Resource Allocation........ 2.5-41
Figure 2.6-1: (U) Representative Sensor Configuration........ 2.6-4
Figure 2.6-2: (U) Representative Flow of Situational Awareness Data 2.6-5
Figure 2.6-3: (U) Vulnerabilities Reported from CERT........ 2.6-59
Figure 2.6-4: (U) Technology Timeline for Network Defense and Situational Awareness...... 2.6-69
Figure 2.7-1: (U//FOUO) Assured Sharing Context Diagram Emphasizing Privileges........ 2.7-6
Figure 2.7-2: (U) Example of Multiple Identities Assigned to a Single User 2.7-18
Figure 2.7-3: (U//FOUO) ECU and Technology Evolution 2.7-33
Figure 2.7-4: (U) Current Key Management Infrastructures........ 2.7-33
Figure 2.7-5: (U//FOUO) KMI – Envisioned Infrastructure 2.7-35
Figure 2.7-6: (U//FOUO) KMI Protected Channel Layers........ 2.7-39
Figure 2.7-7: (U) XKMS Environment........ 2.7-40
Figure 2.7-8: (U) DoD and Commercial Certificate-Managed Infrastructures 2.7-49
Figure 2.7-9: (U) PKI Technology Model........ 2.7-50
Figure 2.7-10: (U) RFID Operation........ 2.7-69
Figure 2.7-11: (U) Audit Life Cycle........ 2.7-82
Figure 2.7-12: (U) Audit Trail Information Flow........ 2.7-84

Figure 2.7-13: (U) Audit Logs – Protection.......... 2.7-86

Figure 2.7-14: (U) Aggregation and Normalization 2.7-88

Figure 2.7-15: (U) Interfaces - Agents and Pipes between Log Devices and the Collection/Monitoring Processes.......... 2.7-89

Figure 2.7-16: (U) Technology Timeline for Assured Resource Allocation.......... 2.7-102

Figure 3.1-1: (U//FOUO) Technology Timeline for Assured Information Sharing.......... 3.1-3

Figure 3.2-1: (U//FOUO) Technology Timeline for Highly Available Enterprise.......... 3.2-11

Figure 3.3-1: (U//FOUO) Technology Timeline for Assure Enterprise Management and Control 3.3-15

Figure 3.4-1: (U//FOUO) Technology Timeline for Cyber Situational Awareness and Network Defense 3.4-24

(U) LIST OF TABLES

Table	Title	Page
Table 1.3-2:	(U) Example of a Technology Adequacy Matrix	2-4
Table 2.1-1:	(U) Hardware Token Standards	2.1-9
Table 2.1-2:	(U) Biometric Standards	2.1-21
Table 2.1-3:	(U) Technology Adequacy for Tokens and Biometrics	2.1-60
Table 2.1-4:	(U) Technology Adequacy for Single Sign-On and Authentication	2.1-61
Table 2.2-1:	(U) Access Control Standards	2.2-11
Table 2.2-2:	(U) Technologies Supporting Access Control	2.2-12
Table 2.2-3:	(U) Minimum Set of IA Attributes for Access Control Decisions	2.2-18
Table 2.2-4:	(U) IC and CES Metadata Working Groups Attribute Comparison	2.2-20
Table 2.2-5:	(U) Metadata Standards	2.2-24
Table 2.2-6:	(U) Metadata Gap Analysis	2.2-27
Table 2.2-7:	(U) Metadata Tool Standards	2.2-35
Table 2.2-8:	(U) Standards on Cryptographic Binding	2.2-39
Table 2.2-9:	(U) Technology Adequacy for Access Control	2.2-46
Table 2.2-10:	(U) Technology Adequacy for Metadata	2.2-48
Table 2.3-1:	(U) Traditional Layered Application Security Standards	2.3-19
Table 2.3-2:	(U) Session Security Standards	2.3-26
Table 2.3-3:	(U) Web Services Security Standards	2.3-31
Table 2.3-4:	(U) FNBDT Standards	2.3-38
Table 2.3-5:	(U) Secure Voice over IP Standards	2.3-57
Table 2.3-6:	(U//FOUO) HAIPE ESP Tunnel Mode Encapsulation	2.3-60
Table 2.3-7:	(U//FOUO) HAIPE State Variable Content	2.3-61
Table 2.3-8:	(U//FOUO) IP Security Technology Readiness Levels	2.3-64
Table 2.3-9:	(U//FOUO) Standards Applicable to IP Security Technology	2.3-66
Table 2.3-10:	(U//FOUO) Standards Applicable to VPN Technology	2.3-73

Table 2.3-11: (U) Secure VoIP Call Control Standards 2.3-84

Table 2.3-12: (U) CDS Standards 2.3-114

Table 2.3-13: (U) Non-Repudiation Standards 2.3-124

Table 2.3-14: (U//FOUO) Secure Voice Technology Gap Analysis 2.3-126

Table 2.3-15: (U//FOUO) Gap Analysis for Non-real-time Application Layer Technologies 2.3-128

Table 2.3-16: (U//FOUO) CDS Technology Gap Assessment 2.3-129

Table 2.4-1: (U) Access Control Standards 2.4-10

Table 2.4-2: (U) Trust Anchor Standards 2.4-13

Table 2.4-3: (U) Policy Language Standards 2.4-20

Table 2.4-4: (U) Distribution Standards 2.4-24

Table 2.4-5: (U) Distribution Security Standards 2.4-28

Table 2.4-6: (U) Directory Standards 2.4-31

Table 2.4-7: (U) Technology Adequacy for Dynamic Policy Management 2.4-33

Table 2.5-1: (U) Technology Adequacy for Assured Resource Allocation 2.5-39

Table 2.6-1: (U) Standards for Intrusion Detection Systems 2.6-33

Table 2.6-2:(U) Network Defense & Situational Awareness Technology Gap Assessment 2.6-64

Table 2.6-3: (U//FOUO) Summary of Technology Gaps 2.6-67

Table 2.7-1 (U) Identity Management Standards 2.7-24

Table 2.7-2: (U) Comparisons of PKI and PMI 2.7-28

Table 2.7-3: (U) Privilege Management Standards 2.7-31

Table 2.7-4: (U) Key Management Standards 2.7-47

Table 2.7-5 (U) Key Management and Certificate Management Standards 2.7-54

Table 2.7-6: (U) Configuration Management Standards 2.7-64

Table 2.7-7: (U) Frequency Ranges for RFID Systems 2.7-70

Table 2.7-8: (U) Inventory Management RFID Standards 2.7-72

Table 2.7-9: (U) Compromise Management Standards 2.7-79

Table 2.7-10: (U) Audit Management Standards 2.7-91

Table 2.7-11: (U) Technology Adequacy for Management of IA Mechanisms and Assets...... 2.7-94
Table A-1: (U//FOUO) Mapping of Technologies to IA System Enablers.................................. A-2
Table B-1: (U//FOUO) TV-1 for IA.. A-6
Table C-1: (U//FOUO) TV-2 for IA.. A-23

369 # (U) REVISION HISTORY

This Table is (U//FOUO)

Revision Number	Description	Date
Revision 1.0 Initial Draft	Initial baseline document that describes the Capability Technology Roadmap. Primary focus is on Identification and Authentication technologies	30 June 2004
Revision 1.0 Final Draft	**General**	15 Oct 2004
	Revised Summary and Executive Summary based on latest technology research and ability to meet Transition Strategy for each Cornerstone	
	Reorganized introduction and eliminated Global Information Grid (GIG) Mission Concept description	
	Added introduction to Section 2 that explains application of TRLs, adequacy levels, and technology timelines in subsequent sections	
	Deleted Appendix on IA Pillars, added appendix with mapping of technologies to section where described, and updated TV-1 and TV-2	
	2.1 Identification and Authentication	
	Refined Enabler description, pulling out Identity Management since it is covered in Enabler 7, Management of IA Mechanisms and Assets	
	Reorganized technologies and add material on Authentication Protocols. Clarified other text	
	Revised gap analysis to reflect adequacy of current technologies to meet 2008 needs	
	Revised recommendations to reflect results of gap analysis	
	2.2 Policy-Based Access Control	
	Editorial Clean-up	
	Expanded Functionality Description	
	Technologies content added and subsection structure changed to reflect results of Technology Analysis Results (Major Technology Subsections: Core RAdAC, Assured Metadata, Digital Access Control Policy)	
	Technologies content added and subsection structure changed to reflect results of Technology Gap Analysis Results (Major Technology Subsections: Core RAdAC, Assured Metadata, Digital Access Control Policy)	
	Section revised to reflect roll-up of Gap Closure recommendation from the major Access Control technology subsections. Also eliminated "Standards," "Technology," and "Infrastructure" subsections as this information subsumed into each major technology subsection	
	2.3 Protection of User Information	
	Added material on Trusted Platforms. Combined (previously empty) sections on Trusted Platforms and Trusted Operating Systems	
	Added material on Trusted Applications. Section was previously empty	
	Added material on Web Security and Application Layer Security	
	Added material on FNBDT and VoIP technologies	
	2.4 Dynamic Policy Management	

	Updated description based upon revised Notional Architecture which is better aligned with RFCs	
	Technologies content added and subsection structure changed to reflect results of Technology Analysis Results (Major Technology Subsections: Development of Policies, Distribution of Policies, Policy Architectures)	
	Technology gap information added to reflect results of Technology Gap Analysis Results (Major Technology Gaps: Expanded policy languages, policy modeling and simulation tools, policy deconfliction tools, and tools or compilers to translate policy language into a device interpretable language)	
	Section revised to reflect roll-up of Gap Closure recommendation from the major Policy Management technology subsections. Also updated timeline for technologies	
	2.5 Assured Resource Allocations	
	Technologies content added (Major Technology Subsections: IA Policy-Based Routing, Operational-Based Resource Allocation, Integrity of Network Fault Monitoring/Recovery and Integrity of Network Management & Control)	
	Technologies content added along with Technology Adequacy matrix	
	Section revised to include Recommendations list and edited Technology Timeline figure	
	2.6 Network Defense and Situational Awareness	
	Protect Technologies content added	
	Situational Awareness: Maturity content added	
	Technologies content added and subsection structure changed to reflect results of Technology Gap Analysis Results (Major Technology Subsections: Core RAdAC, Assured Metadata, Digital Access Control Policy)	
	Intrusion Detection Systems content added	
	Intrusion Prevention Systems content added	
	Cyber Attack Attribution – Editorial clean-up based on comments from John Lowry	
	Correlation Technologies: Technical Detail content added	
	CND Response Actions content added	
	2.7 Management of IA Mechanisms and Assets	
	Key Management - Elaborated on the Maturity Section and the Technology Readiness level (TRL)	
	Audit Management: Modified based on comments and feedback. These had to do with Maturity subsection through Complementary Technologies	
	Added material on Configuration Management of IA Assets, Compromise Management and Inventory Management	
	Modifications and additions were made in order to better represent the Technology Gap Analysis Results	
	Standards, Technology, Infrastructure, and Timelines were all enhanced with additional inputs. The tables were reconfigured, values assigned, and summarized	
	This Table is (U//FOUO)	

370

(U) EXECUTIVE SUMMARY

(U//FOUO) The Office Secretary of Defense/ Networks and Information Integration (OSD/NII) tasked the National Security Agency (NSA) to develop the Information Assurance (IA) component of the Global Information Grid (GIG). This GIG IA Capability/Technology Roadmap document, together with several other documents—including the GIG IA Reference Capability Document (RCD)—describe the IA component of the GIG.

(U) The GIG IA Capability/Technology Roadmap identifies the technologies needed to implement the GIG IA Vision, and it provides a partial evaluation of current and in-development technologies that can or will be able to support GIG needs. As such, the objectives of this document are to:

- (U) Establish, within the context of the GIG IA engineering process, an effective methodology to discover and examine relevant technologies for the purpose of providing guidance to GIG program decision makers and GIG research sponsors
- (U) Provide an assessment of the maturity and suitability of relevant IA technologies to meet GIG IA-required capabilities, focusing in particular on the 2008 milestones of the transition strategy outlined in the GIG IA RCD
- (U) Identify gaps in standards and technologies that will prevent attainment of GIG IA capabilities, and recommend specific actions to take in closing those gaps
- (U) Serve as a means for members of the GIG community and stake holders to gain visibility into the technology roadmap process and provide feedback on appropriate topics, such as standards or significant technologies overlooked during the study to date

(U) In meeting these objectives, this document provides decision makers with the information needed to write new or revise existing standards and policies, develop implementation guidance, make research funding decisions, and devise technology development strategies.

(U) Scope

(U) The GIG IA Capability/Technology Roadmap document presents a fairly complete view of all the technologies that can or should be used to implement IA in the GIG. Those that can support the GIG IA vision are examined in detail. Results are presented to describe the ability of the most promising technologies to fulfill needed GIG IA capabilities in terms of technical capability, maturity, development schedule, and availability. Interdependencies between needed capabilities, technology timelines, and gaps between capability needs and technology availability are also described.

(U//FOUO) In developing the roadmap, the team compared the state, trends, and forecasts of commercial and government technologies available today against the needed capabilities defined in the RCD. Three main categories of information were used. The first is documentation and analyses performed by the NSA as part of development of the IA component of the GIG architecture. This information includes the GIG Mission Concepts, the *As Is* state of GIG programs, and the GIG risk analysis. The second category of information includes current IA standards, technology trends and forecasts available from commercial sources such as Gartner, IDC, etc. and Government trends and forecasts. The third type of information—to be used in subsequent versions of the GIG IA Capability/Technology Roadmap document—is previously-determined technology gaps.

(U) Results

(U//FOUO) The analyses were carried out in the context of the capabilities outlined in the RCD and the Transition |Strategy. In particular, the team assessed the maturity and adequacy of the technologies in meeting the 2008 Vision milestones (Increment I).

(U) The results show that nearly all the Increment I milestones can be achieved if actions are taken immediately to address identified technology or capability gaps. The roadmap provides over 75 specific recommendations to address these gaps. Recommendations range from monitoring ongoing technologies and standards development efforts to ensure compliance with GIG IA needs, to initiating new technology research to support post-2008 milestones) and standards development efforts. We believe that most milestones can be achieved if immediate action is taken on these recommendations.

(U) In our estimation, five milestones defined in the Transition Strategy are unachievable by the specified dates:

- (U//FOUO) *Limited support for end-to-end resource allocation* milestone. Operationally-based resource allocation technologies are very immature, especially considering the constraints and limitations of a secure Black Core. Since there is much research remaining to be done in this area, it is our opinion that a limited capability for allocating resources end-to-end will not be available until 2012—four years after the objective date. The operational impact is a delay in moving from today's *best effort* service for all to efficient resource allocation schemes that ensure priority users receive needed services based on mission criticality to efficient resource allocation schemes.

- (U//FOUO) *All human users identified in accordance with GIG ID standard* milestone. Currently, standards neither exist nor are under development for establishing and maintaining unique, persistent, and non-forgeable identities as will be needed for the GIG. Because of the coordination that will be required across multiple communities, such standards will not likely be in place to support subsequent technology development in time to meet a 2008 objective; however, 2010 is an achievable date for this milestone. No impact on 2008 operational objectives are expected, but delays in meeting 2012 operational objectives is likely. These include: 1) achieving closer collaboration within Communities of

Interest (COI), 2) implementing a global sign-on capability, and 3) achieving Risk-Adaptive Access Control (RAdAC).

- (U//FOUO) *Over-the-network keying for wired and wireless devices* milestone. Efforts are planned for developing the needed security technologies. However an initial capability will not be fielded until 2010, two years after the deadline. Low bandwidth devices, such as wireless nodes, will not be supported until 2012, and coalition networks will not be addressed until 2016. The operational impact is a continued dependence on manual re-keying, which 1) requires greater manpower and costs for handling and safeguarding key material, which will become more troublesome as additional IP encryptors are deployed as the GIG matures, and 2) causes slower response to key compromises, risking more widespread damage.

- (U//FOUO) *Configuration management standards ratified* milestone. Remote configuration products abound, but standards do not yet exist for the secure management of IA-enabled devices. Due to the time needed to draft, coordinate, and achieve consensus among the engineering community, such standards will likely not be ratified before 2008, two years later than the milestone called out in the Transition Strategy. The operational impact is a delay in achieving a consolidated network view. This reduces the overall security posture of the GIG and prevents policy-based network management.

- (U//FOUO) *Audit format and exchange standard ratified* milestone. Auditing products are available today, but the absence of standards, holds-up product integration into the GIG. Developing the needed audit standards and achieving industry acceptance is not likely to be achieved until 2008, two years after the milestone. This will delay the ability to carry-out forensic analysis of attacks and thus hamper computer network defense.

(U) Section 3 provides a summary of the identified gaps and recommendations.

(U) While this version of the document provides the first comprehensive coverage of the technologies, technology assessment is an iterative process: As additional capability needs are identified and IA technologies mature, subsequent analyses will provide recommendations to re-direct current development efforts and initiate new research as needed to meet the GIG visions. These analyses will be documented in subsequent versions of this document, which will be issued on an annual basis.

1 (U) INTRODUCTION

1.1 (U) PURPOSE

(U//FOUO) The GIG IA Capability/Technology Roadmap document is part of the November 2004 deliverables of the Information Assurance (IA) Component of Global Information Grid (GIG) Architecture. Office Secretary of Defense/Networks and Information Integration (OSD/NII) tasked development of the IA component of the GIG architecture to the National Security Agency (NSA).

(U//FOUO) Since the tasking by OSD/NII, the NSA has translated the GIG Vision into derived GIG capabilities and associated IA capabilities. The GIG IA Reference Capability Document (RCD) details the IA derived capabilities by describing the general attributes of each capability. Thresholds and objectives are then defined for each attribute. The thresholds are considered near-term GIG IA requirements to meet the 2008 Vision while the objectives are the capabilities required to meet the GIG 2020 Vision.

(U//FOUO) The GIG IA Capability/Technology Roadmap identifies the current technology trends in IA and compares the trends against the thresholds and objectives identified in the RCD. The result is an availability timeline of anticipated technologies required to support the GIG 2020 Vision.

(U//FOUO) The GIG IA Capability/Technology Roadmap document analyzes the technology trends and technology forecasts (both commercial and government) available today against the capabilities defined in the RCD. The results of the analysis are:

- (U) Capability inter-dependencies
- (U) Technology timelines
- (U) Gaps between capability needs and technology availability

(U//FOUO) The GIG IA Capability/Technology Roadmap document also provides background information and analysis to support decision makers with regard to:

- (U) New/Updated standards
- (U) Infrastructure guidance
- (U) Technology research to fund
- (U) Technology strategy development

1.2 (U) SCOPE

(U//FOUO) Section 2, Information Assurance (IA) System Enabler and Their Technologies, is divided into seven subsections based on the Fundamental System Enablers. Each subsection describes the IA System Enabler, covers the GIG implications of the System Enabler, and describes its related technologies. The related technologies define research areas for technology trends and forecasts to support the development of the technology timelines and the capability/technology gap analysis.

(U) Section 3, Summary, contains a discussion of the technology improvement recommendations needed to meet the Transition Strategy, defined in the RCD, for each Cornerstone. When technologies are missing or unable to meet the strategy, the discussion highlights the impacted operational capability. The four Cornerstones, defined in the GIG IA Operational Concepts Overview document, are:

- (U) Assured Information Sharing
- (U) Highly Available Enterprise
- (U) Assured Enterprise Management and Control
- (U) Cyber Situational Awareness and Network Defense

(U) Section 4 lists acronyms and abbreviations.

(U//FOUO) Appendix A provides a mapping of technologies to IA System Enablers.

(U//FOUO) Appendix B: Technical View (TV)-1 for IA, contains standards that exist today that had not previously been identified as needed to satisfy capabilities listed in the RCD.

(U//FOUO) Appendix C: TV-2 for IA, contains standards that have been identified as needed to satisfy capabilities listed in the RCD but that do not exist today.

1.3 (U) APPROACH

(U//FOUO) The primary guiding principle is to achieve the Objective Goals described in the RCD. This means identifying the necessary technology evolution to fill the gaps between today's IA technology and what is needed for the GIG 2020 Vision's objective system. The IA Risk Assessment helps prioritize—based on security risks—which gaps need to be filled sooner than others. The gap analysis must consider the GIG capability timeline to identify the criticality of each gap.

(U//FOUO) The GIG IA Capability/Technology Roadmap document is built upon three main categories of information. The first category is documentation and analysis performed by the NSA while developing the IA component of the GIG architecture. This information includes the GIG Mission Concepts, *As Is* state of GIG programs, and GIG threats as identified by the GIG Risk Assessment activities. The GIG Mission Concepts capture the NSA's understanding of the capabilities required by the GIG, based on the *To Be* GIG vision as defined by the GIG 2020 architecture and documentation. The *As Is* input captures the IA capabilities currently planned by ongoing GIG programs such as Net-Centric Enterprise Services (NCES), GIG Bandwidth Expansion (GIG-BE), Transformational Satellite (TSAT) Communications, and the Joint Tactical Radio System (JTRS). The GIG Risk Assessment identifies threats to the GIG that must be countered. These threats could be countered in a number of ways, including technology, standards, and policies.

(U//FOUO) The primary document used in development of the GIG IA Capability/Technology Roadmap is the RCD. This includes a description of the GIG Mission Concepts and identifies a set of IA Cornerstones which define the high level IA capabilities required to support the GIG Mission Concepts. This document describes the technologies needed to support the GIG Mission Concepts and IA Cornerstones, but organizes these around the IA System Enablers. The technologies are organized by IA System Enablers because most technologies map to a single Enabler while they are associated with multiple IA Cornerstones. The Summary of this document describes which technologies are needed to support the system capabilities described in the Transition Strategy for each IA Cornerstone. Figure 1.3-1 depicts the GIG Mission Concepts, IA Cornerstones, and IA System Enablers.

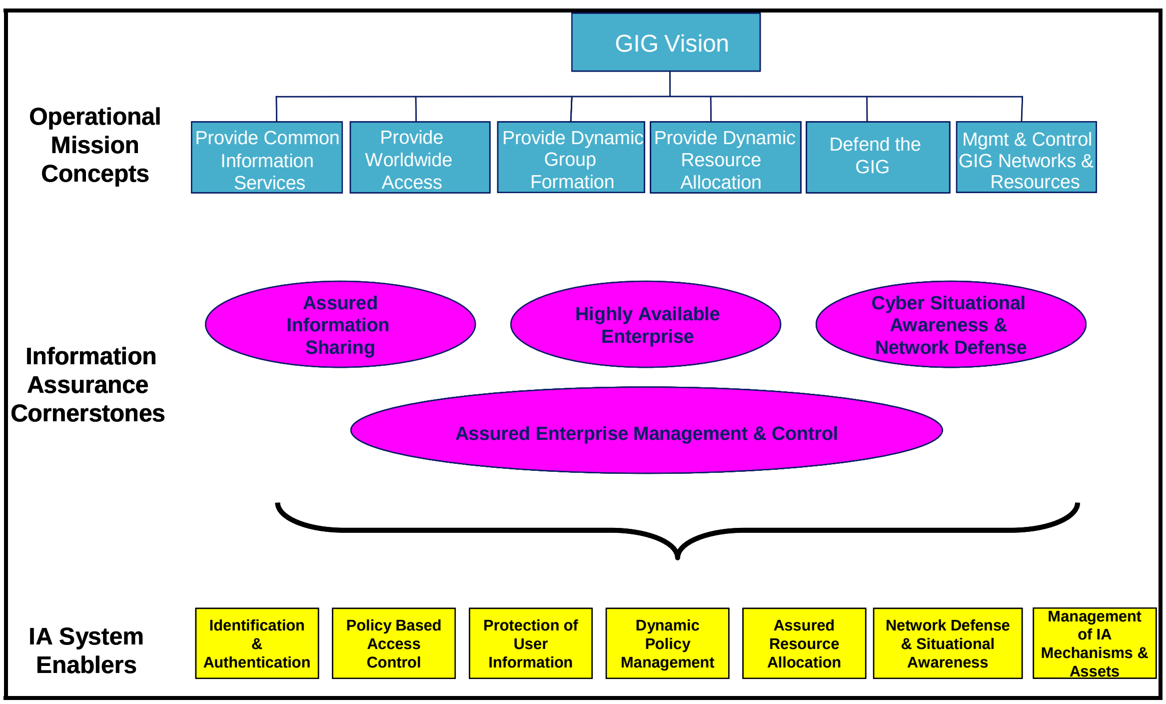


Figure 1.3-1: (U) GIG Mission Concepts, IA Cornerstones, and IA System Enablers

(U//FOUO) The second category of information includes the IA standards in place today, technology trends and forecasts available from commercial sources (i.e., Gartner, IDC), and government trends and forecasts.

(U//FOUO) The third category of information consists of already defined gaps. The expectation is that the process of developing the GIG IA Capability/Technology Roadmap document is an iterative one. And the gaps identified today will drive various activities to close the gaps. These activities could take the form of research, product development, standards implementation, implementation guidance, and policy guidance. The technology development cycle to satisfy a capability could encompass all the previously mentioned forms.

(U//FOUO) The document summary contains an indication of the technology improvement needed to meet the transition strategy—defined in the RCD—for each Cornerstone. When technologies are missing or unable to meet the strategy, the description highlights the impacted operational capability.

(U//FOUO) Any recommendations could be in the form of the need for research, product enhancements, new or enhanced standards, or new or enhanced infrastructure. These recommendations, together with other background information and analysis in this document, are intended to provide decision makers with the information needed for the following decisions:

- (U) Revise or write new standards and policies
- (U) Develop implementation guidance
- (U) Determine direction of research funding
- (U) Devise technology development strategies
- (U) Develop technology implementation plans

(U//FOUO) Figure 1.3-2 graphically represents the iterative development of the GIG IA Capability/Technology Roadmap.

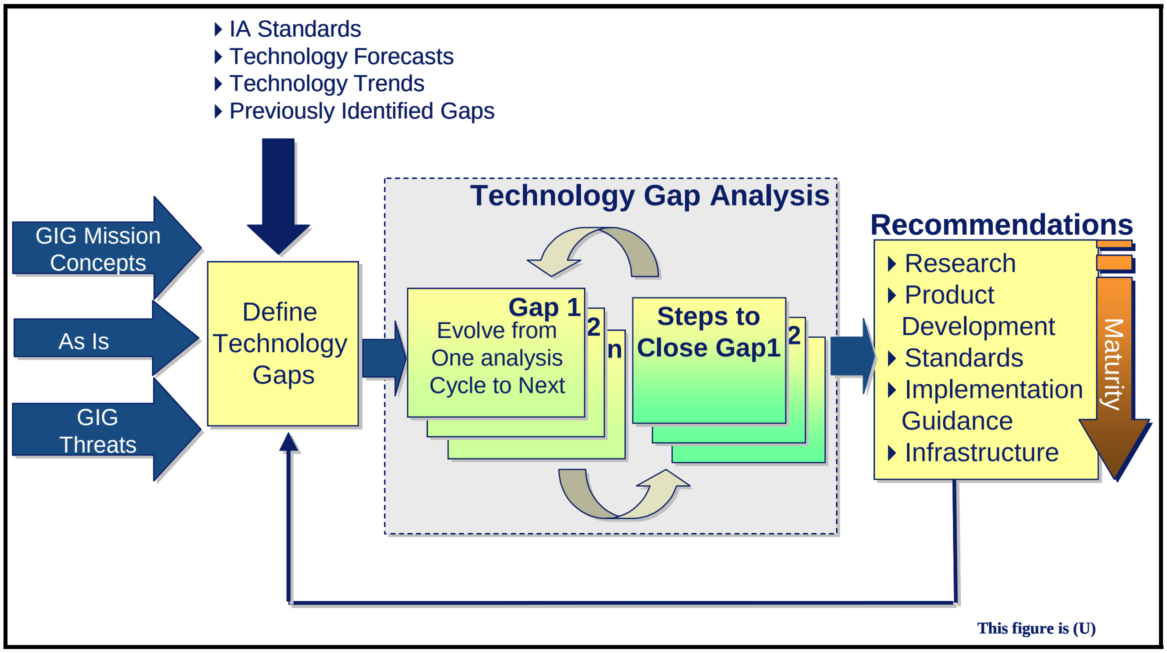


Figure 1.3-2: (U) Iterative Development of the GIG IA Capability/Technology Roadmap

(U) As a technology progresses through the development cycle, the current state of the development is input into the analysis process. The result of the analysis could be the closing of a gap or the identification of additional gaps between capabilities and the technology.

(U) The work of this document is an iterative process that will require re-analyses as additional capability needs are identified and technologies mature. Future releases of this document will be issued on an annual basis.

2 (U) IA SYSTEM ENABLERS AND THEIR TECHNOLOGIES

(U//FOUO) Information assurance (IA) is essential to meet the six GIG Mission Concepts. Without IA, the GIG would not only fail to provide the right information, at the right time, at the right place, in support of warfighting, business, enterprise information environment and national intelligence, but the GIG could also be a haven for other nation states, cyber terrorists, hackers, and malicious insiders to further disrupt operations in support of national objectives.

(U//FOUO) While a large body of technologies exist to at least partially provide the IA capabilities stipulated by the GIG IA Vision, evolution of most existing technologies is needed, and new technologies must be invented. This section of the document identifies the needed IA technologies—both existing and to be developed. Preliminary assessments of technology maturity and identified gaps are presented, which should aid decision makers in guiding existing and starting new technology development efforts.

(U) For convenience of analysis and organization, the IA technologies are categorized and presented in the context of the IA Enablers[1] they support. This "binning" into IA Enablers ensures minimal technology overlap and complete coverage of the needed IA capabilities to better support technical gap analysis. The IA Enablers used to organize the subsequent subsections are:

- (U) Identification and Authentication
- (U) Policy-Based Access Control
- (U) Protection of User Information
- (U) Dynamic Policy Management
- (U) Assured Resource Allocation
- (U) Network Defense and Situational Awareness
- (U) Management of IA Mechanisms and Assets

(U//FOUO) Each subsection presents all aspects and benefits of the associated IA Enabler. The IA Enabler itself is described, and key features of the IA Enabler are defined. An overview of the supporting types of technologies is presented, organized around the IA Enabler. Each technology is described in sufficient detail to support gap analysis. Finally, results of the technology gap analyses are presented, and technology development timelines and recommendations for the IA Enabler are provided. The technology timelines, showing the date that each technology will be available for integration into the GIG, are optimistic; they are based on ideal circumstances, e.g., adequate funding and appropriate technical manpower are available to begin and execute the recommended research, or existing developments continue as currently planned. Adverse budgetary decisions will obviously delay the availability of the technologies for use.

[1] (U) The seven IA Enablers are core constructs that, together, provide the IA component of the GIG. These serve as architectural building blocks for enabling the GIG Mission Concepts.

(U//FOUO) A fairly comprehensive description is presented for each technology, but only to the extent needed to support recommendations for subsequent development efforts. Details of specific product implementations are avoided where possible to avoid conferring vendor endorsement, which distracts from the purpose of this analysis. Rather, numerous technology implementations were researched and considered, and only one or a small number were selected for inclusion in the roadmap, according to authors' opinions of how well these implementations represent the state of practice. In this context, specific items included for each technology are as follows:

- (U) Technical details: Description of the technology in terms of technical characteristics, features, and theory of operation. Consistent with the goals of the roadmap, the description may cover the superset of capabilities represented by the combination of a few related implementations or products.

- (U) Usage considerations: Discussion of potential implementation issues peculiar to the technology and anticipated operating environments, advantages of the technologies, and risks—in terms of potential threats and attacks—that might be incurred in employing the technology.

- (U) Maturity: Description of the current state relative to the goal capability of the technology itself. (This is not to be confused with the GIG IA capability that the technology would support.) While it is desirable to specify maturity of every technology in terms of Technology Readiness Level[2] (TRL), the roadmap does not attempt to do so, because either a TRL could not be found and there is insufficient information on which to base a specific estimate, or the analysis is based on multiple products/implementations that are each at different stages of development. Instead, then, the overall development stage of each technology is assessed and described by one of three maturity level terms:

  - *(U) Early* refers to technologies that are in the research or analysis phase (corresponding to TRL range 1-3).

  - *(U) Emerging* refers to those in the initial prototyping and lab demonstration phase (TRL range 4-6).

  - *(U) Mature* refers to technologies that are undergoing operational demonstration, production, and deployed operations (TRL range 7-9).

(U) In addition, specific TRLs are provided where they could be determined with a fair degree of certainty.

- (U) Standards: Discussion of standards that are pertinent to the technology. Included are existing standards, or those that will need to be developed in order to support the technology.

- (U) Costs/limitations: Discussion of the costs and limitations the technology would pose on the GIG architecture and connected systems when they are significant. Examples are

[2] (U) There are nine TRLs defined in Appendix 6 of DoD Instruction 5000.2, ranging from basic principals observed and reported (level 1) to actual system proven through successful mission operations (level 9).

where the technology would impose significant operational manpower burden (amount, caliber, and training), extraordinary procurement costs, undue complexity, unusual integration difficulties, adverse or significant impact on warfighting operations, or significant communications bandwidth or processing overhead.

- (U) Dependencies: List of related items, such as other technologies and data, upon which the technology must depend in order to provide the described capability.
- (U) Alternatives: List of possible alternative technologies or techniques that could support the IA Enabler, either for early adoption to provide an interim capability, or as a substitute if the described technology does not mature.
- (U) Complementary techniques: List of additional technologies or techniques that improve or enhance the described technology.

(U//FOUO) To facilitate discussions of the gap analyses, one or more matrices are provided for each IA Enabler. These are intended to summarize the explanations and show, at a glance, how adequately the analyzed technologies meet the capabilities defined by the IA Enabler for the 2008 GIG implementation (Increment 1). Technologies are combined into categories for simplification. The adequacy level, determined by how well the sum of the assessed technologies in each category addresses each IA Enabler attribute, is described in Table 1.3-1. Capability attributes from the RCD are included in the matrices for reference.

Table 1.3-1: (U) Definitions of Technology Adequacy Levels

This Table is (U)

Adequacy Level	Indication	Definition
Not Applicable	N/A	There is no expectation that the technology category could support the IA Enabler attribute.
Unknown	White	Technology investigation not completed, e.g., no result presented
Completely uncovered	Light gray	No technology is available, and no research is underway to develop the needed technology(ies), to address the IA Enabler attribute
Some coverage, but insufficient	Light black/white grid	R&D is underway that should lead eventually to at least partially covering the IA Enabler attribute, and anticipate that the resulting technology will be available in time to meet GIG IA milestone dates, *OR* A technology exists in the category that partially meets the needs of the IA Enabler attribute now, but additional technology R&D is needed to either enhance it or add to it in order to fully satisfy the attribute, *OR* Taken together, a combination of existing products or technologies in the group could satisfy the IA Enabler attribute now, but additional work is needed to combine the technologies in order to fully satisfy the attribute.
Fully adequate	Solid black	Technology, or a compatible combination of technologies, is available now that fully meets all aspects of the IA Enabler attribute, *OR* Technology development is underway and on schedule to fully satisfy the attribute at the time needed.

This Table is (U)

(U//FOUO) Table 1.3-2 shows an example technology adequacy matrix for the Policy Based Access Control Enabler. Here, Digital Rights technologies are needed only to support the Object Life Cycle and Protection Profile attributes. This is indicated in the table by the black grid and gray shading under Digital Rights technologies under the Object Life Cycle and Protection Profile IA attributes and N/As under the Digital Rights technologies for all other IA attributes. Access Control Policy technologies are needed to support all IA Attributes, as shown by the black grid and gray shading. The white intersection of Access Control Policy technology and the Protection Profiles attribute indicates technologies are neither available nor research underway to satisfy the Protection Profiles attribute.

(U//FOUO) A matrix filled with black and "n/a" entries would reflect the ideal situation where all IA attributes are satisfied with technologies.

Table 1.3-2: (U) Example of a Technology Adequacy Matrix

This Table is (U)

		Technology Categories			
		Core Access Control	**Digital Rights**	**Access Control Policy**	**Required Capability (RCD attribute)**
IA Attributes	**Risk & Need Determination**		N/A		IAAC4
	Math model		N/A		IAAC4
	Decision logic		N/A		IAAC1, IAAC4, IAAC7
	Ontology	N/A	N/A		IAAC4
	Exception handling		N/A		IAAC5
	Conflict resolution		N/A		
	Object Lifecycle				IAAC8
	Protection Profile				IAAC9

This Table is (U)

2.1 (U) IDENTIFICATION AND AUTHENTICATION

(U//FOUO) I&A mechanisms provide critical IA foundations toward achieving the GIG Vision of assured information sharing. In the assured sharing model, information is exchanged among entities (e.g., individuals, devices) on the enterprise infrastructure. Similarly, services are shared among entities on the enterprise infrastructure.

(U//FOUO) Access to information or services is based upon several factors including entity properties, their authentication mechanism, properties of the objects to be accessed, the IT components, the environment in which the entities exist, and the access control policy implemented. All of this is based on the ability to uniquely identify the entities participating in the exchange and the authentication mechanisms used by the entities participating in the transaction. The ultimate goal is to support a SSO process independent of the many roles and privileges of the entities involved.

(U//FOUO) The Identity and Authentication (I&A) Enabler is the sum of the mechanisms and processes that result in a composite level of trust of an entity that can be used in all access control decisions on the GIG during a given service request or login session. Entities that need to be identified and authenticated include human users, workstations, networks, services, and other resources.

(U//FOUO) The level of trust of an entity is referred to as its I&A Strength of Mechanism (SoM) Score. Each service request is examined to determine how resistant the authentication of that request is to impersonation or forgery. The likelihood that a service request was forged depends on both the difficulty of forging the request, as measured by the I&A SoM, and the motivation and ability of the adversary.

(U//FOUO) To support I&A SoM scoring on the GIG it is necessary to develop the following:

- (U//FOUO) Standards and technical requirements for assigning assurance levels for each of the following factors that affect I&A strength and for deriving the I&A SoM score from those factors:
  - (U//FOUO) Strength (resistance to compromise) of identity proofing during user registration
  - (U//FOUO) Strength of the user’s authentication token
  - (U//FOUO) Strength of the protocols used to authenticate service requests
  - (U//FOUO) Strength of the user’s operating environment (e.g. clients, IT components, and network).
- (U//FOUO) Mechanisms for conveying to services the assurance level of a specific service request and of the IT components used to generate and process the request.
- (U//FOUO) Policies that make use of I&A SoM scores and other assurance measures in the decision to grant or deny access to particular resources

2.1.1 (U) GIG Benefits due to I&A

(U//FOUO) The Information Assurance constructs used to support I&A provide the following services to the GIG:

- (U//FOUO) Provides assurance that every entity participating in a GIG transaction is who he/she/it claims
- (U//FOUO) Enables accountability for all GIG actions
- (U//FOUO) Accommodates varying trust levels for users and IT components by identifying how much an entity can be trusted
- (U//FOUO) Enables capability for single sign-on (SSO) once the identity is recognized and trusted throughout the GIG

2.1.2 (U) I&A: Description

(U//FOUO) Unique identity and identity proofing are fundamental to the I&A process. Unique IDs are created for all entities (e.g. individuals, devices, services). Identity proofing refers to the methods used to prove an individual's or devices identity before issuing a Unique ID. Identity proofing mechanisms for individuals could range from providing no proof of identity presented to requiring multiple picture IDs be presented in person by the individual receiving the Unique ID. The identity registered for an individual is unique and remains constant despite changes of that individual's name or other attributes. More information on Identity Management is provided in Section 2.7, Management of IA Mechanisms and Assets

(U//FOUO) The authentication mechanism used in conjunction with this ID is also critical to granting access to shared data and resources. The strength of the authentication mechanism measures resistance to attempts to guess, sniff, extract, or otherwise compromise the entity's authentication material. Current authentication mechanisms for human individuals include:

- (U) User ID and password
- (U) Use of software PKI certificates to provide a verifiable identity
- (U) Use of a Hardware Token that contains an entities PKI certificate and on-board mechanisms to verify an entity's identity
- (U) Biometrics to unlock a token that protects the non-forgeable PKI certificate containing the identity that is shared in a protected manner during authentication exchanges

(U//FOUO) The User Profile is a logical collection of information associated with a user, but it is not necessarily stored in a single location. The identity management system must store a basic user record containing the unique ID and the core identifying information that was verified (e.g., birth certificate information, driver's license number) or collected (biometrics) during identity proofing. Other information that is logically part of the user profile must be strongly bound to the user's unique ID but may be stored separately. For example, public key certificates used to authenticate the user may be stored in a hardware token or certificate repository, role information may be stored as signed attributes in a privilege server, contact information may be stored in a user directory, and subscription information may be stored in a discovery server.

(U//FOUO) After registration, a user may log into a GIG asset using the authentication token issued to that user. At the conclusion of the login process, an authenticated session would exist, which has an associated I&A SoM session score. Authentication information for service requests can either be derived from the user's login session or generated by the user's token for each request. When the service provider can directly authenticate the user's original request, the request assurance score can be determined directly based on the user and client assurance. But in architectures where requests are passed through multiple providers, each of which can authenticate only the preceding requestor, the originator's assurance score is decreased at each intermediate processing point. In either case, the final request assurance score is determined based upon the following factors:

- (U//FOUO) Identity Proofing method used to register the user
- (U//FOUO) Token used to authenticate the user's identity (e.g., password, software certificate, hardware certificate, biometrics)
- (U//FOUO) Authentication mechanism used for the request or session (e.g., unbound identity assertion, Secure Session Layer (SSL) session, signed request)
- (U//FOUO) The properties of the device used to logon (based upon their configuration and management of the devices some devices may not be as trusted as others) and each device in a trust chain between the originator and the provider
- (U//FOUO) The user's location or operating environment (e.g., highly trusted network, remote access via a computer on the Internet)

(U//FOUO) Some participants in the GIG may require the entity and session to be periodically re-authenticated. For example, if the data being retrieved is critical mission data, then the data sharer may want to re-validate that the parameters of the original session login are still valid. This may entail a requirement for the data requestor to provide their biometric data periodically to ensure they are still present.

2.1.3 (U) I & A: Technologies

(U//FOUO) The following technology areas support the Identification and Authentication Enabler:

- (U) Authentication Tokens
- (U) Biometrics
- (U) Device/Service Authentication
- (U) Authentication Protocols
- (U) Authentication Confidence
- (U) Single Sign-On (SSO)

(U) The three basic means of user authentication (and what they are based upon) are:

- (U) Authentication by knowledge (what a user knows, e.g., a fixed memorized password)
- (U) Authentication by characteristic (what a user is, e.g., a biometric)
- (U) Authentication by ownership (what a user has, e.g., a token)

(U) The main disadvantage of fixed passwords is that they are vulnerable to various attacks, including social engineering, sniffing (e.g., network and/or electromagnetic emanations), dictionary attacks, maliciously planted Trojan-horse software, etc. Once a user's fixed password is compromised, it is impossible to detect subsequent system accesses by malicious parties.

(U) Section 2.1.3.1 discusses the token technologies that support authentication by ownership. Biometric technologies are discussed in Section 2.1.3.2.

2.1.3.1 (U) Authentication Tokens

2.1.3.1.1 (U) Technical Detail

(U) Authentication tokens provide a means for a user to dynamically generate a single-use one-time password (OTP) every time a remote secure system is accessed. This thus avoids fixed password vulnerabilities. Tokens may be implemented in either hardware or software.

(U) A hardware token is a device, which the user in some manner employs to interface (either physically or indirectly through user interaction) with a local client processor (e.g., a PC), requiring secure access to a remote server processor or system. This hardware token contains the critical security parameters for the authentication process.

(U) A software token is implemented within the local client processor itself and thus depends upon the security and trustworthiness of the client's operating system. Examples of standard OTP authentication protocols that are functional equivalents of software tokens include S/Key, OPIE (One-Time Passwords in Everything, http://inner.net/opie), and SSH (Secure Shell).

(U) Most implementations of authentication tokens require the user to enter a PIN (personal identification number) to locally unlock the token functionality (and thus are not subject to network sniffing attacks). A PIN can be viewed as a primitive fixed and memorized password. A biometric also can be used to unlock token functionality. This combination of independent authentication factors provides a stronger authentication mechanism and prevents system compromise if a hardware token is lost or stolen.

(U) Tokens function by using either Symmetric Key Authentication (a single shared secret key known at both the client and server) or Public Key Authentication (where the client knows only the private key, and the server knows the public key). All authentication tokens work by producing dynamic single-use OTPs based upon credentials unique to the issued user and upon a cryptographic algorithm or hash function. Symmetric Key Authentication and Public Key Authentication are further explained in Section 2.1.3.4 which describes authentication protocols in general.

(U) There are several basic token authentication modes under symmetric key, grouped within the categories of Asynchronous and Synchronous.

2.1.3.1.1.1 (U) Asynchronous Token Authentication Mode

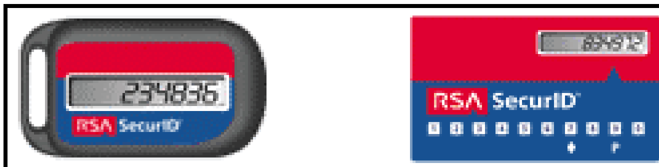
(U) Challenge-Response: In this mode, the user sends his username to the server in order to identify the shared secret key. The server generates a random challenge and sends it back to the user. The user keys the challenge into the token. This challenge is then cryptographically processed with the secret key in order to generate a response. The response is then entered onto the client and sent back to the server. The server independently does the same process and compares results.

(U) This mode is 'asynchronous' because there is no (time-based) requirement that the response arrive at the server within a prescribed and limited amount of time, nor is the response a function of any underlying event counter.

2.1.3.1.1.2 (U) Synchronous Token Authentication Mode

2.1.3.1.1.2.1 (U) Time-driven

(U) In this mode, both user token and server generate an OTP based upon the shared secret key and an internal (network-synchronized) clock value. In order to permit network transmission time variations, the clock value resolution may be on the order of 60 seconds or less (to allow for clock drift). An example of this token type is SecurID by RSA. The user reads the varying time-based OTP from the LCD display of the hardware token (See Figure 2.1-1—Note the option for a 10-digit numeric keypad for entry of PIN to enable the token). The user then inputs this number onto the client processor, and it is sent to the server where it is compared with the server's expected value. A match yields successful authentication.



This is figure is (U)

Figure 2.1-1: (U) Examples of time-driven hardware tokens

2.1.3.1.1.2.2 (U) Event-driven

(U) In this mode, instead of using time to create an OTP, an authentication event counter value is used with the shared secret key to generate the one-time password.

(U) Time and event driven modes are examples of response-only authentication schemes since the process only requires one-way transmission from user to server. A third version of response-only authentication is accomplished by both the user token and server. This creates a response from a hidden random challenge (rather than a mere time or counter value, which could be viewed as more predictable and certainly as monotonically increasing). This challenge is derived from the previous challenge, which is ultimately derived from some random seed value created at token initialization and known to both token and server.

(U) Authentication modes could be combined (e.g., challenge-response + time-driven + event-driven) to provide stronger authentication, just as stronger authentication is achieved by combinations of independent authentication factors (password/PIN + one or more biometrics + token), the.

2.1.3.1.2 (U) Usage Considerations

2.1.3.1.2.1 (U) Implementation Issues

(U) Hardware tokens are implemented in a variety of physical form factors. Those that require only indirect user-interaction (i.e., user observation of displays on the token, followed by manual entry of data onto the local client processor) include pocket-style calculators and key fobs. Hardware tokens that connect directly to the client processor include *smart* cards and Universal Serial Bus (USB) tokens. An example of smart card authentication tokens is the DoD Common Access Card (CAC), which can also be used as a photo ID card and for physical access control.

2.1.3.1.2.2 (U) Advantages

(U) In general, authentication tokens have the basic advantage that they can be used over public networks and are not subject to compromise by hostile network sniffing, since the authentication information is cryptographically based and unpredictable (i.e., not subject to standard replay attacks).

(U) Hardware tokens are inherently resistant to social engineering attacks, since it is very unlikely that an innocent user would provide an attacker with both the token and its enabling PIN. Another obvious distinct advantage of hardware tokens is their portability, which enhances the user's mobility and capability to authenticate remotely by home PCs, laptops, or personal digital assistants (PDA).

(U) Smart cards (and USB tokens), since they interface directly with a user client, offer the advantages that they can be used as a safe repository for sensitive personal data, such as PKI credentials, passwords, and various account numbers. Smart cards have onboard processors, which can do critical authentication processing (e.g., generating a cryptographic digital signature) without being observed by a potential attacker (as opposed to the alternative of doing the processing on a client processor, which may have been compromised by Trojan horse software). Protection of sensitive information on the smart card when it is not being used is accomplished by tamper-resistant—both physical and electronic—encryption of any stored data and the required use of an enabling PIN.

2.1.3.1.2.3 (U) Risks/Threats/Attacks

(U) A basic disadvantage or risk of hardware tokens is that they can be lost or stolen. However, in the case of smart cards such as the DoD CAC, the privileges of a lost card can be revoked or canceled by the centralized PKI infrastructure authority. In addition, unless the enabling PIN is also known by the malicious possessor of a lost/stolen token, that token can not be used in a compromising manner.

(U) In the deployment of any authentication token system, especially in the case of an organization like the DoD with large numbers of geographically dispersed users, secure token distribution requires a robust proof of delivery (POD) mechanism (e.g., by manual signature for non-repudiation).

(U) Public key authentication systems also suffer from potential risks if they have weak public key management or certification. These systems rely on the clear and verifiable binding between a user identity and the associated public key by the public key certificate. Only a trustworthy and reliable certification/registration authority can assure that the certificate database is valid and up to date.

(U) Another potential risk is that a hardware token can be left enabled at a client workstation, which could allow a malicious intruder to masquerade as the valid user. A potential solution to this might be to require periodic biometric checking/re-authentication.

(U) Besides the risk of potential attack where a hardware authentication token is physically taken from the valid user—through loss, theft, or misplacement—there are further potential attacks on the authentication process *at a distance* from the actual token itself. The classic attack would be the *man-in-the-middle* attack against the collaborative process between the remote user and the centralized authentication server. In this case, the attacker would have access to the communications path somewhere on the network between the communicating parties. A man-in-the-middle could inject, delete, or alter data that is sent in either direction. However, due to the unpredictable cryptographically-based nature of the authentication responses sent by the user to a server, it is unlikely that a man-in-the-middle could predict a future authentication value response and thus could not gain access to the system.

(U) Another attack that could be mounted *at a distance* from the hardware token would be planting malicious attacker Trojan horse modifications in the client workstation. This could be partially avoided by having the authentication process done primarily within the token itself and not allowing the shared symmetric secret key, or private key, to ever be transmitted off the token. Finally, a guaranty that this secret key is safe can be made if some form of physical tamper resistance is built into the token itself. Such tamper resistance would also prevent alterations to any software that operates on the token itself.

(U) Of course, a token and its associated authentication function can be assumed to be secure only if the main system authentication server is non-hostile and has not been compromised in any manner. Thus, since the server is potentially the worst location for single point failure, the most effort should be expended in safeguarding this resource.

2.1.3.1.3 (U) Maturity

(U) Authentication token technology has matured significantly, especially when each sub-technology is viewed as an independent component. Current and future work needs to be done in integrating the sub-technologies, along with the complementary authentication enhancing technologies such as biometrics. An example of this is the DoD CAC with added biometric functionality. In summary, the Technology Readiness Level of tokens can be thought of as Mature (TRL 7 -9).

2.1.3.1.4 (U) Standards

(U) There are a variety of standards arenas—both formalized and actual—which play a role in the development and evolution of authentication tokens and their underlying protocols and algorithms.

2.1.3.1.4.1 (U) Hardware Token Standards:

(U) Organizations and arenas responsible for developing standards related to smart card technology and other tokens include RSA Labs Public-Key Cryptography Standards (PKCS), Microsoft Crypto API (CAPI), Personal Computer/Smart Card (PC/SC), and the ISO International Organization for Standardization. These standards are listed in Table 2.1-1

Table 2.1-1: (U) Hardware Token Standards

This Table is (U)	
Name	**Description**
RSA Labs PKCS Standards	
PKCS #11	Cryptographic Token Interface (*cryptoki*) Standard (specification of an application programming interface API for cryptographic token devices)
PKCS #12	Personal Information Exchange Syntax (specifies transfer syntax for personal identity information such as private keys and certificates, etc.)
PKCS #15	Cryptographic Token Information Format Standard (ensures interoperability of multiple vendor implementations)
Microsoft API Standards	
CAPI	Cryptographic Application Programming Interface standards
PC/SC Standards	
PC/SC Workgroup Specifications 1.0	Interoperability Specs for Smart Cards and PCs (platform and OS independent)
PC/SC Workgroup Specifications 2.0	Updated enhancements, including contactless (wireless RF) cards
ISO Standards	
ISO/IEC 7810	Identification Cards – physical characteristics
ISO/IEC 7811	ID Cards – Recording techniques
ISO/IEC 7812	ID Cards – Identification of issuers
ISO/IEC 7813	Financial transaction cards
ISO/IEC 7816	ID Cards with contacts
ISO/IEC 10373	ID Cards – Test Methods
ISO/IEC 10536	Contactless ID Cards – Close Coupled
ISO/IEC 14443	Contactless ID Cards – Proximity (*Mifare* cards) - 1-inch range
ISO/IEC 15693	Contactless ID Cards – Vicinity (*I.CODE* cards) - 5-inch range
This Table is (U)	

(U) The RSA PKCS specifications originated in the early 1990s from RSA Labs and, though from a single company (versus a collaborative standards body), have been subsumed into and adopted by numerous de facto and formalized standards. The PC/SC specifications are both PC platform and PC operating system independent, while also specifying low-level device interfaces. The updated version is addressing contactless/wireless smart card specifications. The ISO smart card standards are derived from the basic ISO identification card standards. Similar to the RSA PKCS #11, Microsoft's CAPI for Windows defines application programming interface (API) for accessing tokens and letting vendors integrate security products into the OS—without token developers having to write separate drivers for each application.

2.1.3.1.4.2 (U) DoD Common Access Card

(U) An emerging *standardized* authentication token within the Department of Defense is the DoD Common Access Card (CAC), an example of which is shown in Figure 2.1-2:

This is figure is (U)

Figure 2.1-2: (U) DoD Common Access Card

(U) The characteristics of the DoD CAC will define a de facto smart card standard, merely by virtue of the vast number of CACs that will eventually be issued (to reserve and active military, DoD civilian employees, and DoD contractors on DoD networks such as the emerging GIG).

(U) The main directed requirements of the CAC were that it provide for encryption of secure messages, digital signature for non-repudiation, hardware token capability for storage of cryptographic keys for use on unclassified networks, and flexible smart card technology to support the efficient evolution of DoD identity-based business processes. Additional requirements were that a CAC work as a photographic identification card, provide for facility physical access control, provide logical access (with strong Identification authentication) to unclassified DoD networks, and take advantage of the existing card-issuance infrastructure. Hardware token smart cards are a solution that satisfies all of these requirements.

(U) In order to support legacy applications, the CAC includes both bar codes and magnetic stripes. Current work is being done to include contactless/wireless versions of the CAC (for easy facility/physical access applications).

(U) The CAC is a credit-card sized smart card that conforms to ISO/IEC standards 7810 and 7816. The basic processor CPU on the current version card is an 8-bit microcontroller (with newer versions having up to 32-bit RISC reduced instruction set processors), the memory is 32K EEPROM (plans for 64K), and the operating system is a Java API application programming interface (to allow for a multiple vendor open architecture). Crypto processing is done by an 1100-bit, advanced crypto engine (and a 112-bit/192-bit DDES-ECC crypto accelerator). Double DES (DDES) is used instead of single DES, which was originally used in many commercial token implementations. CAC vendors include Gemplus, Axalto, and Oberthur. The DoD implanted 3 Java applets on the card—the PIN security applet, the PKI applet, and a generic personal information management applet. DoD is currently looking at adding an enhanced biometric (e.g., fingerprint/thumbprint) capability directly onto the CAC.

(U) Issuance of a new CAC to a DoD employee requires a user fingerprint template collection and verification and self-selection of an enabling PIN. At this time, about 82 percent of the over 4.4 million potential CAC recipients have been issued their cards (with cards being issued at up to 12,000 a day). The CAC issuance infrastructure includes about 1,600 stations at more than 900 sites around the world.

(U) The DoD also plans to develop a central issuance facility. In order to complement the issuance of CACs, the DoD has already purchased more than 2 million stand-alone card readers for use with existing PC computers (with new PCs being purchased with embedded card readers).

(U) Due to the extremely large number of DoD CACs being distributed and due to their application in sensitive areas and operations, much thought is going into the development and evolution of the CAC. Thus it can be viewed as a robust de facto implementation standard of a smart card token. It and its descendants will be important tokens in the future GIG.

2.1.3.1.5 (U) Cost/Limitations

(U) The cost and limitations of authentication tokens is based on both the token functionality itself and upon the required supporting infrastructure—both local (as in the case of requiring peripheral card readers for smart card tokens) and centralized (as in the case of a PKI Public Key Infrastructure with its associated Certificate Authority [CA]).

(U) The concept of a PKI is very straightforward, but in order to implement a PKI that adaptively scales to support a large user population, large investments must be made and complexities overcome. One cost advantage of the DoD CAC smart card is that, by serving multiple legacy functions, it will enable the DoD to eliminate and phase out many legacy identity cards and thereby provide a larger than might be expected return on investment.

(U) Symmetric single-key software tokens implemented on a user client PC are significantly cheaper than the equivalent hardware token implementation, since there is no hardware cost beyond the already existing PC. An imputed lower mental cost to the user is that much of the authentication process is hidden from the user. Another cost of software tokens is the lack of operating system independence.

(U) Depending on how complex (and inherently costly) one wishes to make either smart cards or USB tokens when doing public key authentication, one can tradeoff the processing demands placed upon the token device and the host client. In the cheapest and simplest token, it can simply act as the repository of the private key that it can export to the client, which would then do any required cryptographic processing. The low monetary cost of this approach however incurs potentially high security risks (and costs) since the client PC must now be fully trusted to be impervious to malicious attacks (i.e., Trojan horses).

(U) The alternate to this approach is to do cryptographic processing on the token itself (as on the DoD CAC). This may be done by either first generating the needed private key on a client workstation and then storing it on the token (Off-Token Key Generation) or by generating the private key only on the token itself (On-Token Key Generation).

(U) The cost/limitation of Off-Token Key Generation is that it may temporarily expose the private key to potential hacking attacks on the client (although another advantage is that the user can make a backup copy of the key for disaster recovery purposes). The potential cost or limitation of On-Token private key generation is that if the token suffers a hardware failure, the private key may be lost forever. An example of a vendor product that does on-token key generation is the cryptographic smart card by Datakey Inc.

(U) Despite their ease of security and convenience in carrying on one’s key-chain, the fact that they can do onboard processing and storage, and the prevalence of USB ports on client computers, there are several inherent limitations and costs to USB authentication key-fob tokens. USB ports are often very inconveniently located on PCs (i.e., on the back panel of a PC tower). USB ports may not be physically robust enough to avoid being damaged by the repeated daily (or more often) interface with a USB token. And finally, a USB token is not large enough to easily incorporate a photo ID.

(U) The DoD CAC has several current limitations. It was developed originally for use on the NIPRNet and not on systems that require higher assurance. For example, the CAC is not NIAP evaluated (specifically, a High Assurance Protection Profile does not currently exist), and it contains foreign COTS hardware and software (e.g., one of the vendors is Gemplus, a French manufacturer).

(U) The GIG will require higher assurance tokens that provide a way to present identity credentials and authentication for access to classified information, which is an option not currently supported by the CAC. The three primary Java security applets (access control/PIN security, PKI support, generic information container management) need to undergo full high assurance security evaluation.

(U) Plans are also being made to utilize asymmetric (public key) cryptography for the purpose of transport of keys and integrate this capability into the CAC issuing system by December, 2006. The January 2008 goal, to deliver a new DoD CAC compliant high assurance token that is manufactured only in the U.S with only U.S.-developed software, will provide a CAC that delivers high assurance Identification Management capabilities for the full suite of GIG customers including DoD, the Intelligence Community, and International Partners. This high assurance token will be able to carry classified information and Type I keying material.

2.1.3.1.6 (U) Dependencies

(U) Further evolution of the DoD CAC to include full biometric integration and contactless/wireless RF capability will rely on the full developing and maturing of the PC/SC Workgroup Specifications 2.0. Biometric integration also depends upon acceptance of a biometric technique (e.g., fingerprint).

2.1.3.1.7 (U) Alternatives

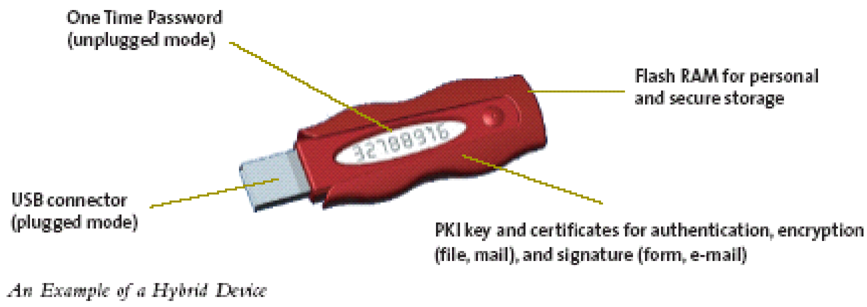
(U) The basic stand-alone alternatives to tokens (what you have) are biometrics (what you are) and simple fixed passwords (what you know).

2.1.3.1.8 (U) Complementary Techniques

(U) Though biometrics and simple passwords (or PINs) can be viewed as mere alternatives to authentication tokens, they are better viewed as adjuncts or complementary techniques that when combined together have a multiplicative effect on an overall system security. Biometric data templates can be stored securely on a smart card token rather than on the client workstation. There is even the possibility of integrating an actual biometric fingerprint/thumbprint reader onto the surface of a smart card, thus eliminating the need for additional peripheral hardware.

(U) The concept of an all in-one security device is an example where complementary techniques are combined. Security devices can embed many, if not all, base authentication methods. The intent is to create highly flexible and versatile security devices, such as for authentication, encryption, signing, secure storage, and physical access. Comprehensive functionality and personalization (e.g., personal storage) are essential to encourage users to embrace security devices such as a token on a key chain or a smart card in a wallet. By supporting multiple strong authentication methods, the same device becomes capable of interacting with a wide range of networks and applications.

(U) The remote access scenario shows the benefit of integrating multiple authentication methods into one single security device. Figure 2.1-3 shows a USB token with either a PKI-enabled SIM chip inside or a smart card, with a display integrated within the reader to display the OTP. With this hybrid device, a user roams over a Wi-Fi network using SIM-based authentication. Once on the public network, the user can initiate a virtual private network (VPN) connection to a gateway using the RSA private key and certificate, which are stored in the token. Once the VPN tunnel is established, the user can log on to a portal to access the user’s account through a Web interface—using the One Time Password generated by the token.



This is figure is (U)

Figure 2.1-3: (U) Example of a Hybrid Device

(U) An additional complementary technique would be mere physical access controls to secure facilities. Though this serves more as a member of an authorized group identification authentication than as an individual identification authentication, it is an important first barrier that must be overcome before a malicious intruder can get anywhere close to sensitive IT equipment. Indeed, the DoD CAC serves the dual purposes of both facility access (with both the photo ID feature and legacy magnetic stripe for card swipe-controlled physical accesses) and the follow-on required identification authentication for use of sensitive IT network resources. Other physical access control technologies are being researched that use facial scans to enable access to computer resources. An advantage to this approach is that it is a continual authentication, so each time the user leaves the computer locks the user's screen. When the user returns to the computer, the facial recognition authentication is repeated to re-authenticate access.

2.1.3.1.9 (U) References

(U) "One Time Passwords in Everything", http://inner.net/opie, by Craig Metz.

(U) "PKCS Public-Key Cryptography Standards", http://www.rsasecurity.com/rsalabs/node.asp?id=2124 (RSA Labs).

(U) PCSC Workgroup, http://www.pcscworkgroup.com/

(U) "Multi-Biometric Verification Demonstration (Category: Secure Access to Physical Systems, Devices and Spaces)", Vijayakumar Bhagavatula, Dept of ECE, Carnegie Mellon.

2.1.3.2 (U) Biometrics

2.1.3.2.1 (U) Technical Detail

(U) A biometric is a measurable, physical characteristic or personal behavioral trait that can be used to recognize the identity—or verify the claimed identity—of an enrollee.

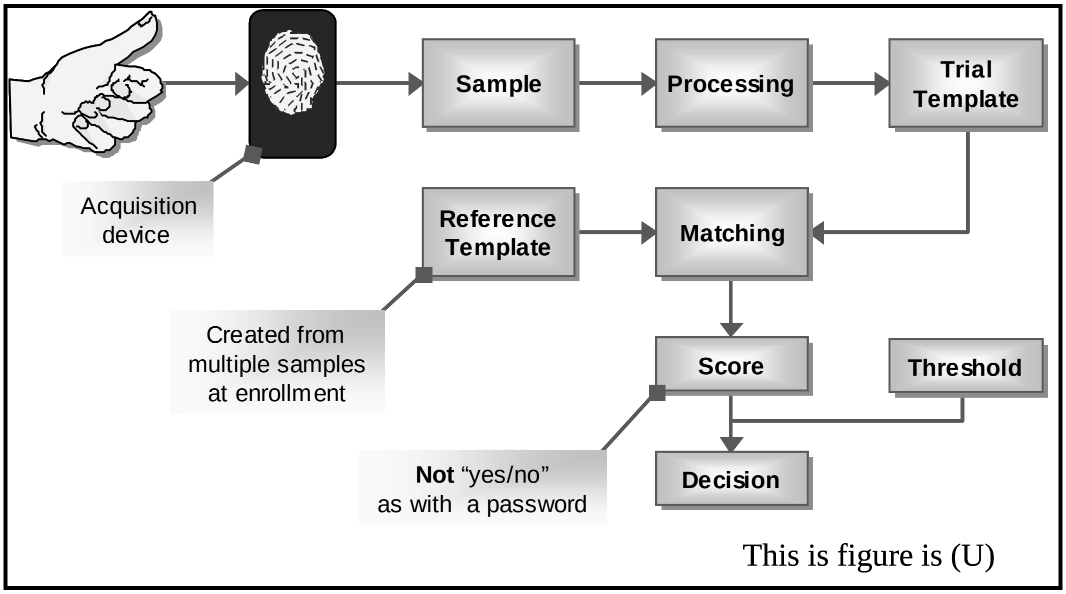


Figure 2.1-4: (U) Biometric System Block Diagram

(U) Two processes are necessary for any biometrics system: enrollment and verification. Enrollment involves recording the user's biometric and storing it in the system as a template. Verification is the comparison of a user's biometric against the reference template to verify a user's identity. The enrollment process typically happens during system initialization or when a new user is added to the system.

(U) Biometric systems all perform the same basic process for verification, as illustrated in Figure 2.1-4. First a biometric acquisition device reads the user's biometric and creates a trial template. A template is data that represents the biometric measurement of an enrollee used by a biometric system for comparison against previously or subsequently submitted biometric samples. The trial template is then compared against a reference template, previously stored during the enrollment process.

(U) If biometrics is used with other authentication factors, the reference template for the user's claimed identity can be retrieved and compared against the trial template to verify the user's identity; this is referred to as an authentication mode. If a biometric is the only authentication factor, the trial template must be compared against all reference templates in the database until a match is found; this is referred to as a recognition mode. The matching process is based on a scoring system. The system must judge whether there is a close enough match between the trial and reference templates.

(U) The accuracy of a system is measured by its False Match Rate (FMR) and False Non-Match Rate (FNMR). The FMR is the probability that the biometric system will incorrectly identify an individual or will fail to reject an impostor. The FNMR is the probability that the biometric system will fail to identify an enrollee or verify the legitimate claimed identity of an enrollee. Generally the lower the FNMR the easier the system is to use while the lower the FMR, the better the security of the system. These characteristics are typically configurable by an administrator. For a biometric system that uses just one template matching attempt to decide acceptance, FMR is the same as False Acceptance Rate (FAR). When multiple attempts are combined in some manner to decide acceptance, FAR is more meaningful at the system level than FMR. For a biometric system that uses just one attempt to decide acceptance, FNMR is the same as False Rejection Rate (FRR). When multiple attempts are combined in some manner to decide acceptance, FRR is more meaningful at the system level than FNMR.

(U) During enrollment some biometric systems perform multiple scans of the same biometric to create the reference template. This can create a more accurate reference template and help reduce the FMR and FNMR.

(U) Accuracy is also driven by the amount of data collected or the number of data points collected in the reference sample. This also contributes to storage requirements: more data points means more storage capacity is required, which translates into more cost.

(U) Reliability is affected by aging and environmental conditions. Injuries and background noise could affect the accuracy of the devices and increase the FNMR.

(U) There are many biometric factors that can be used. They are generally broken down into two categories: physiological and behavioral. Physiological biometrics is usually derived from a person's anatomy and are difficult to alter. Examples include fingerprints, iris, and hand print. Behavioral biometrics are derived from an action performed by an individual. Behavioral biometrics are usually easier to alter but can be perceived as less intrusive by the user. Examples of behavioral biometrics include signature, voice recognition, and gait.

2.1.3.2.1.1 (U) Physiological Biometrics

2.1.3.2.1.1.1 (U) Fingerprint Recognition

(U) The patterns of friction ridges and valleys on an individual's fingertips are unique to that individual. For decades, law enforcement has been classifying and determining identity by matching key points of ridge endings and bifurcations. Fingerprints are unique for each finger of a person including identical twins.

(U) Fingerprint recognition is the most widely available biometric technology. Fingerprint recognition devices for desktop and laptop access are now widely available at a low cost from many different vendors. With these devices, users no longer need to type passwords—instead; only a touch provides instant access. Fingerprint systems can also be used in identification mode. Several states check fingerprints for new applicants to social services benefits to ensure recipients do not fraudulently obtain benefits under fake names.

2.1.3.2.1.1.2 (U) Face Recognition

(U) The identification of a person by the facial image can be done in a number of different ways. It can be done by capturing an image of the face in the visible spectrum, using an inexpensive camera, or by using the infrared patterns of facial heat emission. Facial recognition in visible light typically models key features from the central portion of a facial image. Using a wide assortment of cameras, the visible light systems extract features from the captured image(s) that do not change over time while avoiding superficial features such as facial expressions or hair. Several approaches to modeling facial images in the visible spectrum are Principal Component Analysis, Local Feature Analysis, neural networks, elastic graph theory, and multi-resolution analysis. The major benefits of facial recognition are that it is non-intrusive, hands-free, continuous, and is acceptable to most users.

(U) Some of the challenges of facial recognition in the visual spectrum include reducing the impact of variable lighting and detecting a mask or photograph. Some facial recognition systems may require a stationary or posed user in order to capture the image, although many systems use a real-time process to detect a person's head and locate the face automatically.

2.1.3.2.1.1.3 (U) Iris Recognition

(U) The iris of the eye is the colored area that surrounds the pupil. Iris patterns are unique. The iris patterns are obtained through a video-based image acquisition system. Iris scanning devices have been used in personal authentication applications for several years. Systems based on iris recognition have substantially decreased in price, and this trend is expected to continue.

(U) The technology works well in both verification and identification modes (in systems performing one-to-many searches in a database). Current systems can be used even in the presence of eyeglasses and contact lenses. The technology is not intrusive. It does not require physical contact with a scanner. Iris recognition has been demonstrated to work with individuals from different ethnic groups and nationalities.

2.1.3.2.1.1.4 (U) Hand and Finger Geometry

(U) These methods of personal authentication are well established. Hand recognition has been available for over twenty years. To achieve personal authentication, a system might measure physical characteristics of either the fingers or the hands. These include length, width, thickness, and surface area of the hand. One interesting characteristic is that some systems require only a small biometric sample (a few bytes).

(U) Hand geometry has gained acceptance in a range of applications. It can frequently be found in physical access control in commercial and residential applications, in time and attendance systems, and in general personal authentication applications.

2.1.3.2.1.2 (U) Behavioral Biometrics

2.1.3.2.1.2.1 (U) Signature Verification

(U) The technology is based on measuring the speed, pressure, and angle used by the person when a signature is produced. One focus for this technology has been e-business applications and other applications where signature is an accepted method of personal authentication.

2.1.3.2.1.2.2 (U) Speaker Recognition

(U) Speaker recognition has a history dating back some four decades, where the output of several analog filters was averaged over time for voice matching. Speaker recognition uses the acoustic features of speech that have been found to differ between individuals. These acoustic patterns reflect both anatomy (e.g., size and shape of the throat and mouth) and learned behavioral patterns (e.g., voice pitch, speaking style). This incorporation of learned patterns into the voice templates (the latter called voiceprints) has earned speaker recognition its classification as a behavioral biometric.

(U) Ambient noise levels can impede collection of the initial and subsequent voice samples. Performance degradation can result from changes in behavioral attributes of the voice and from enrollment using one telephone and verification on another telephone. Voice changes due to aging also need to be addressed by recognition systems. Many companies market speaker recognition engines, often as part of large voice processing, control, and switching systems. Capture of this biometric is seen as non-invasive. By using existing microphones and voice-transmission technology to allow recognition over long distances by ordinary telephones (wire line or wireless), this technology needs little additional hardware.

2.1.3.2.2 (U) Usage Considerations

(U) There are two typical implementations for deploying a biometric system: using a centralized database for storing user reference biometric templates (recognition mode) or storing the biometric value directly on a token the user possesses (authentication mode).

2.1.3.2.2.1 (U) Implementation Issues

(U) Recognition mode uses a centralized database containing all enrolled users' reference templates. A user presents himself/herself at the biometric reader for authentication. The reader collects the biometric, digitizes it, and sends it over the network from the client (directly connected to the reader) to a Biometric Authentication Database. The comparison and acceptance/rejection of the fingerprint/face/etc. is made there, and the acceptance or rejection notice is sent back to the client. If a match is verified, the user is allowed to access the various resources on the network.

(U) Authentication mode typically stores the biometric value directly on the user's token. In this case there is no central database. Rather, the user feeds a hardware token into the reader, and then presents the fingerprint, face, etc., for reading. The reader is a trusted device that compares the measured biometric directly with the value stored on the presented token.

(U) Biometrics may not be suitable for every environment. For example, users in tactical environments may have difficulty using a fingerprint reader since their fingers might get dirty or cut or their protective clothing may preclude access to the biometrics reader. Carrying a large biometric reader with a handheld device may limit the device's mobility. Hence, use of a particular biometric must be weighed against its operational environment. The authentication confidence associated with biometrics must consider the applicability of the authentication mechanism for the environment in question.

2.1.3.2.2.2 (U) Advantages

(U) The time required to perform a match in authentication mode is much less than in recognition mode because the trial template must only be matched against a single reference template. The time necessary to perform the recognition process is driven by the size of the template database and the size of the template. The more users enrolled in a system the longer it will take to perform a match in the database. Also the larger the template the longer a positive match will take. Using biometrics as one of several authentication factors increases the strength of the authentication, and because the biometric system can be used in authentication mode versus recognition mode, it should not impact system performance.

(U) To access some information in the GIG, multifactor authentication may be required. Biometrics can play an important role in providing a higher authentication score than a simple user name and password. They can also be used to unlock a user's privileges or other authentication information. Biometrics also assist in providing an audit function as they can uniquely identify a user and enable the system to tie the user to performed actions.

2.1.3.2.2.3 (U) Risks/Threats/Attacks

(U) With the recognition mode implementation, an adversary does not need to attack the reader, but rather the network or the biometric database. The biometric template must be secure as it crosses the network. If the template can be captured, an adversary can present it to the biometric database and impersonate an authorized user. This can be avoided by securing the connection between client and database by using protocols such as IPsec or TLS, which includes replay protection.

(U) The database itself also is a target for attack. If the database can be compromised, all reference templates stored on it are also compromised. The database is likely to be riding on an OS that can be exploited through a variety of methods, much like attackers on the Internet capture credit card databases today. Alternatively, an attacker can use the weakness to replace the stored value with his own value, thus granting him access while completely eliminating the legitimate user from the system.

(U) The difference between this biometric attack and credit card attacks is that biometric templates are very difficult to revoke. If an attacker captures a set of credit card numbers, those cards can be revoked and new cards issued. Or if an attacker captures a set of private encryption keys from a PKI, the certificates corresponding to those keys can be revoked and new keys/certificates issued. While there is some pain and expense in the revocation operation, the procedures and methods are known.

(U) Contrast this with an attack that captures the digital fingerprints of the user base. The attacker now has the digitized fingerprints and can inject them into the system as needed to impersonate a user. It is not practical to have users get new fingerprints; the only option is to throw out the existing biometric solution and replace it with a new one (e.g., a new method of digitizing fingerprints that bears no relation to the other one and cannot be derived from it or switch to using face recognition instead of fingerprints).

(U) To defend against these attacks, a number of steps must be taken:

- (U) The digitized image must be some transform of the actual biometric that cannot easily be reversed. For example, the value sent, stored and compared would be a SHA-1 hash of the digitized fingerprint. If this were to be captured, it would be replaced with a SHA-2 hash of the face, etc.

- (U) Each use of the biometric should include some unique value (e.g., time stamp) hashed in with the actual value to protect against replay attacks. This is a trade-off, as the goal would be to use a biometric value for an entire session (e.g., only capture the fingerprint once, then let the user work for a few hours), and replay attacks can potentially be done whenever the time is still within the legal window of use of the biometric.

- (U) As indicated above, the communication between the computer connected to the biometric reader and the central database must be secured, for example, using TLS, IPsec, or equivalent security.

- (U) The computer on which the Central Database resides must be secured to the maximum extent possible.

- (U) Protected Resources must also be secured. They must be able to authenticate all accesses by users—they should be able to tell from where an access arrives in case it is attempted by an attacker who has compromised the system.

(U) The authentication mode implementation avoids the network and operating system-based vulnerabilities described above; however, it presents a number of its own potential vulnerabilities. Chiefly, these relate to the tamper resistance of the hardware token—if an attacker can acquire the token and replace the stored value with his own value, he will be approved by the system.

(U) Other vulnerabilities with this approach relate to how the biometric reader communicates successful matching to the system. If an attacker can simply forge a successful match message from the reader to the protected resources, the attacker is in the system again.

2.1.3.2.3 (U) Maturity

(U) The Gartner Hype Cycle lists two to five years to reach the plateau/adoption. The plateau is defined as “the real-world benefits of the technology are demonstrated and accepted.” Gartner lists several factors, which determine the maturity level. User acceptance is one of the primary factors along with ease of use, accuracy, reliability, resistance to attack, and cost.

(U) User acceptance is a concern with iris and retina scanning, because of a general fear people have about instruments close to their eye. The accuracy of iris and retina scanning is reasonably good, but the cost is high for scanning equipment. Voice and signature recognition are neither as intrusive as iris and retina scanning nor as expensive, but are not as accurate and require more effort to use. Fingerprint, face, and hand recognition fall in between in terms of intrusiveness, accuracy, and expense.

(U) IDC lists three main challenges to adoption of biometrics authentication: convenience, installation, and portability. Convenience translates into ease of use in Gartner's terms while installation is really a cost factor, which includes time and money. Portability is something Gartner does not discuss.

(U) IDC describes portability as how easy is the biometric device to carry around. If the biometric device is cumbersome to carry, people will refuse to use it.

(U) Gartner lists the following obstacles to biometrics technology:

- (U) Biometric equipment is expensive to buy and install
- (U) Applications have to be changed
- (U) None of the biometrics devices are fool proof
- (U) Accuracy can be affected by aging, injury, or environmental conditions

(U) There are several initiatives that may accelerate the biometric development market. For example, a trusted traveler program is being lobbied for to move people through airports quickly and to improve security. One of the fundamental pieces to a trusted traveler program is biometrics. Travelers must be authenticated as they move through the transportation system. While a trusted traveler program is still being debated in Congress, a pilot program is underway. Developments related to the trusted traveler program could accelerate the biometrics market.

(U) When it comes to the core algorithms and mechanisms involved, the Technology Readiness Level of biometric technologies in general can be thought of as nearing the Mature level (TRL7-9).

2.1.3.2.4 (U) Standards

(U) Standards applicable to biometrics are listed in Table 2.1-2.

Table 2.1-2: (U) Biometric Standards

This Table is (U)	
Standard	**Description**
Common Biometric Exchange Formats Framework (CBEFF)	CBEFF originally stood for Common Biometric Exchange File Format and was originally developed by the Biometric Consortium (BC). It was published by NIST as NISTR 6529. CBEFF defines a standard method for identifying and carrying biometric data. It describes a framework for defining data formats that facilitate the communication of biometric data. CBEFF does not specify the actual encoding of data (e.g., bits on a wire) but provides rules and requirements and the structure for defining those explicit data format specifications.
This Table is (U)	

This Table is (U)

Standard	Description
BioAPI	The BioAPI standard defines an Application Program Interface (API) and a Service Provider Interface (SPI) for standardizing the interaction between biometric-enabled applications and biometric sensor devices. The API provides a common method for applications to access biometric authentication technology without requiring application developers to have biometric expertise. The SPI allows the production of multiple BSPs (Biometric Service Providers) that may be used by an application without modification of that application, regardless of biometric technology. The BioAPI Consortium originally developed the BioAPI specification. The BioAPI Consortium is a group of over 50 organizations focused solely on furthering a standard biometric API. M1 has taken the resulting specification from the consortium and standardized it nationally as ANSI INCITS 358-2002. M1 has also contributed ANSI INCITS 358-2002 to SC 37 where it is currently a draft international standard.
Data Interchange Formats	A data interchange format specifies the low-level format for storing, recording, and transmitting biometric information. This biometric information may be unique to each biometric characteristic (e.g., fingerprint, iris, signature) and/or to each method of capture (e.g., photograph, capacitive sensor). In some technologies, this biometric information is called a *template*. M1.3 is currently working on projects dedicated to standards for the following formats.
Biometric Profiles	A biometric profile identifies a set of base biometric standards that apply to a single application or scenario. The profile then identifies the appropriate configurations, parameters, and choices for options provided within those specifications. The goal is to provide interoperability and consistent functionality and security across a defined environment. M1.4 is engaged in the following projects: • Interoperability and Data Interchange—Biometric Based Verification and Identification of Transportation Workers • Interoperability, Data Interchange and Data Integrity—Biometric Based Personal Identification for Border Management • Point-of-Sale Biometric Verification/Identification SC 37 has defined a functional architecture that serves as part one of a multi-part standard. SC 37 is also working on the first profile of the standard titled *Biometric Profile for Employees*.
Biometric Evaluation Methodology	The Biometric Evaluation Methodology (BEM), Version 1.0, was designed to aid security evaluators who were attempting to evaluate biometric products against the Common Criteria (CC). The Common Evaluation Methodology (CEM) used in CC evaluations does not address the environmental, user population, and other issues that have an impact on a biometric implementation. The BEM specifically addresses these issues as they apply to biometric technology evaluations under the CC. Evaluators, certifiers and developers from Canada, U.K., GERMANY, U.S., Italy, Sweden, and others developed the BEM. Version 1.0 of BEM was released in August of 2002.

This Table is (U)

This Table is (U)

Standard	Description
Biometrics Protection Profile	The CC is an effort of the US, Canada, and European countries to establish a common set of security criteria by which to evaluate IT products. This effort has resulted in an international standard (ISO/IEC 15408-1) for evaluating IT security products. The document that establishes the implementation-independent security requirements for a given category of product is called a Protection Profile. Currently, the DoD Biometrics Management Office (BMO) and the National Security Agency (NSA) are developing four Protection Profiles for biometrics products: • Robustness Biometric PP for Verification Mode • Basic Robustness Biometric PP for Verification Mode • Medium Robustness Biometric PP for Identification Mode • Basic Robustness Biometric PP for Identification Mode
Biometric API for JavaCard	The JavaCard Forum was established in 1997 to promote Java as the preferred programming language for multiple-application smart cards. A subset of the Java programming language was proposed for these cards and resulted in a standard for a JavaCard API. The JavaCard Forum has extended the JavaCard API to enroll and manage biometric data securely and facilitate a match on card capability with the Biometric API for JavaCard. The Biometric API manages templates, which are stored only in the card. During a match process, no sensitive information is sent off the card.
Common Data Security Architecture (CDSA), Human Recognition Services Module	The Human Recognition Services Module (HRS) is an extension of the Open Group's Common Data Security Architecture (CDSA). CDSA is a set of layered security services and a cryptographic framework that provides the infrastructure for creating cross-platform, interoperable, security-enabled applications for client-server environments. The biometric component of the CDSA's HRS is used in conjunction with other security modules (i.e., cryptographic, digital certificates, and data libraries) and is compatible with the BioAPI specification and CBEFF.

This Table is (U)

2.1.3.2.5 (U) Cost/Limitations

(U) Biometrics can provide an enhanced authentication capability but they have several costs associated with them. First, biometric readers must be deployed on the system. This may be a substantial cost depending on the cost per reader and the number of readers required. In the GIG, it is envisioned that many systems will require biometric authentication, and therefore a large number of readers will be required.

(U) There are several processes that require administration in a biometric system and therefore add to the maintenance cost of the system. One of these processes is enrollment, which incurs a cost both upon the central administrator and upon the user.

(U) Another limitation of biometrics is the user's acceptance. This is influenced by the perceived intrusiveness of the biometric. For example, signatures are widely accepted today, and a user would be far less likely to mind a signature biometric than an iris or retina scan that requires them to put their eye close to the biometric reader. If the users will not accept the use of the particular biometric technology, it cannot be expected to be successful.

2.1.3.2.6 (U) Alternatives

(U) Alternatives for biometrics include any information that can be used to verify a user's identity. For example, Government issued photo identification may be substituted for a biometric for applications such as physical access to a building. However, it alone is not adequate to authenticate access to an information system.

(U) Another alternative to biometrics is to require more information that the user knows. For example, if a biometric is not available, inputting several passwords may be sufficient to authenticate the user.

2.1.3.2.7 (U) Complementary Techniques

(U) Hardware tokens are complementary to biometric implementations using the authentication mode.

2.1.3.2.8 (U) References

(U) Biometric Authentication Perspective (Gartner)

(U) Hype Cycle for Information Security, 2003 (Gartner)(U) "Reduced Complexity Face Recognition using Advanced Correlation Filters and Fourier Subspace Methods for Biometric Applications", by M. Savvides, PhD Thesis, May 2004, Electrical & Computer Eng, Carnegie Mellon University

2.1.3.3 (U) Device/Service Authentication

(U//FOUO) Security and trust in any network is a function of all the elements that make up a network. This includes end-point (client and server) devices that can impersonate users and organizations. As network devices proliferate (e.g., mobile phones, PDAs, portable digital music players, set-top boxes, and laptops), the ability to distinguish between trusted and rogue devices becomes a fundamental security requirement.

(U) Since an authenticated device can act as the root of trust, it can also provide the security foundation for a new breed of applications, such as identity based anti-virus solutions and digital information rights management software. From this standpoint, device and service authentication is a core requirement of any strong identification management strategy.

(U) There are a variety of initiatives and incentives/motivations that are driving industry towards robust device authentication, including the following:

- (U) Transform today's mobile devices (e.g., cell phones, PDAs, laptops) into strong authentication devices
- (U) Propagate device credentials, strong authentication algorithms, and authentication client software across many network end points (e.g., desktop computers, servers, switches, Wi-Fi access points, set-top boxes)
- (U//FOUO) Enhance device credentialing management schemes for improving SSO in the GIG, or at least to help reduce Sign-On problems
- (U) Build around well-established infrastructure components such as directory and RADIUS servers
- (U) Proliferate low-cost, multi-function authentication devices (e.g., tokens, smart cards)
- (U) Facilitate native support (e.g., platform connectors) for strong device and user authentication in application development and identification management platforms
- (U) Leverage federated identity protocols as a powerful propagation and integration mechanism
- (U) Enable best-of-breed solutions through interoperable components
- (U) Credentials and Security Devices

2.1.3.3.1 (U) Technical Detail

2.1.3.3.1.1 (U) Universal Strong Authentication for Devices

(U) The strength—the trustworthiness—of an identity depends on multiple factors. The initial authentication process (i.e., identity verification), the type of credential being issued (i.e., security token), and the depth of the relationship between the authenticator and the authenticated entity all contribute to the strength of an identity. Beyond the authentication process, the security policies enforced by the authentication authority and its operation best practices have a direct impact as well.

(U) Strong identification management must take into account technology, policy, and operational issues. Strong authentication is the first level of trusted networks where identities can be securely shared and trusted across independent partners. It is the foundation for a more secure network, one in which all people and all devices are strongly authenticated in an open, interoperable, and federated environment.

(U) Three methods specify the core types of authentication credentials—SIM secret and X.509 certificate. Each of these methods has a specific use in an interoperable environment:

- (U) SIM-based authentication – SIM (Subscriber Identity Module). This authentication method predominates in telecommunications. It also is emerging as an important authentication method in public Wi-Fi networks (authentication and roaming across Global System for Mobile Communications/General Packet Radio Service and 802.11 networks).

- (U) PKI-based authentication – PKI is a fundamental security component of all major Internet protocols for authentication and communication (e.g., Transport Layer Security [TLS], WS-Security, IPsec IKE, 802.1x, Session Initiation Protocol [SIP]). The choice of X.509v3 certificates as strong credentials is also consistent with deployment trends in enterprise and government markets. Furthermore, certificates offer additional security functionality beyond authentication, for example for electronic form and e-mail signing and file encryption. It should also be noted that there are ongoing developments within PKI/KMI to specify not just devices in the Directory Information Tree, but also services, servers and roles.

2.1.3.3.2 (U) Usage Considerations

(U) When describing authenticating a device, it is important to consider to what the device is authenticating. In the case of 802.1x, the device is being authenticated at the link layer. In the case of a call setup on a mobile phone network, the authentication occurs at an application level. Sometimes authentication will need to be done on a per connection basis (such as on a point-to-point link). Other times, authentication will need to be done at an enterprise level for auditability and scalability purposes.

(U) Each of these different scenarios implies a different mechanism to perform device authentication. This can lead to many overlapping (and potentially conflicting) protocols and processes.

2.1.3.3.2.1 (U) Advantages

(U) Secure device authentication enables many other security goals of GIG-related technologies. By also authenticating a device that a user is interacting with, the entire system has a higher degree of confidence in the authenticated session. By authenticating a device in a data center communicating with another unmanned device, services such as web services can use the identity of a device as a foundation for trust in the end-to-end system. Device authentication permits secure access to networks, applications, and any other GIG-connected resources.

2.1.3.3.2.2 (U) Risks/Threats/Attacks

(U) Device authentication mechanisms have many potential points of vulnerability. The protocol used to relay authentication across the network may be a point of attack. Dr. Arbaugh from the University of Maryland has already found several weaknesses in the 802.1x protocol. These vulnerabilities allow 802.1x to be attacked over the network. These attacks may allow an attacker to either hijack a session from an authenticated device or prevent a legitimate device from using the network.

(U) Furthermore, device authentication may be relying on the physical security of the device itself. This security may come in the form of guards, guns, and dogs (standard physical security) or may be the result of the use of tamperproof/tamper evident devices such as a smart card. The guards, guns, and dogs model of physical security can be overcome by physical force. Tamperproof/tamper evident protections might be overcome by sophisticated technical attacks. Ross Anderson has published many papers on the topics of subverting tamper resistant/ proof devices.

(U) However the device authentication mechanism is subverted, the end result is generally the same; lack of trust in the actual identity of the end device. When designing or deploying device authentication systems, great care must be exercised to determine the real security limitations of the protocols and products involved.

2.1.3.3.3 (U) Maturity

(U) Device authentication is an emerging technology. Until recently, there has been little perceived value in authenticating a device. Enterprises have been more worried about the identity of the user and have not focused their attention on the device itself. However, as devices become more mobile and disposable, device authentication is rapidly gaining visibility.

(U) Unfortunately, few standards exist and even fewer products. This area of device authentication still requires a great deal of research and standards development before widespread market adoption will occur.

(U) In summary, the Technology Readiness Level of device authentication can be viewed as Emerging (TRL 4 – 6).

2.1.3.3.4 (U) Standards

2.1.3.3.4.1 (U) 802.1x

(U) The Institute of Electrical and Electronics Engineers (IEEE) approved the standard 802.1x on June 14, 2001. This standard is based on the physical characteristics and identification of the device, port, or wireless station that is requesting the connection. The standard provides a mechanism for restricting access to a local area network (LAN) or a virtual local area network (VLAN). Generally, it is described as providing port-based access control.

(U) The 802.1x authentication architecture consists of a supplicant—a user or entity representing the endpoint requesting a network connection; an authenticator—a network device or entity that is facilitating the authentication of the supplicant; and an authentication server or service—responsible for validating the supplicant's credentials and determining whether to authorize the authenticator to grant access to the requested services.

(U) 802.1x specifies how to carry link-level authentication information using Extensible Authentication Protocol (EAP). (See the next section.) While 802.1x does not require the use of a separate authentication service, it is often deployed in combination with a RADIUS server.

2.1.3.3.4.2 (U) EAP

(U) EAP, or Internet Engineering Task Force (IETF) RFC 2284, is an authentication framework that defines a way to encapsulate different authentication methods. EAP can be used in combination with point-to-point protocol (PPP) (IETF RFC 1661) or IEEE 802.1x. A recent Internet draft updates the original EAP specification.

(U) A range of methods have emerged that build on EAP, including:

- (U) EAP-Transport Layer Security (TLS), for encrypted communication between endpoints identified by public key infrastructure (PKI) certificates
- (U) EAP-message digest 5 (MD5), for password authentication using a challenge-response approach
- (U) EAP-Generic Token Card (GTC), for use with one-time password tokens
- (U) EAP-Microsoft Challenge Handshake Authentication Protocol (MS-CHAPv2)

(U) Cisco, Microsoft, and RSA collaborated in proposing Protected EAP (PEAP) to the IETF. PEAP has security improvements that extend Cisco's Lightweight EAP (LEAP). LEAP uses a stronger password-hashing authentication approach than EAP-MD5, but is also susceptible to offline dictionary attacks against the password. PEAP is supported by Microsoft, Cisco, Funk Software, and Meetinghouse Communications, but is not recognized as an industry-wide standard. Typically, PEAP is used in combination with TLS for secure communication between endpoints that are authenticated using a method other than PKI.

2.1.3.3.4.3 (U) RADIUS

(U) RADIUS, most recently specified by IETF RFC 2865, was originally designed as a protocol mechanism for authenticating remote users. It is still typically used today to authenticate remote users connecting to a dial-in modem pool or an Internet-accessible, virtual private network (VPN), gateway device.

(U) The typical architecture for RADIUS involves the VPN gateway or access server acting as the client, requesting authentication of a user connection; and a RADIUS server, performing the authentication and passing back appropriate configuration information to the requesting service. In addition, RADIUS servers can act as proxies for other RADIUS servers or authentication services. This is often required when users are roaming between service providers or interfacing between a service provider and an internal network's identification management infrastructure.

(U) While RADIUS is independent of 802.1x, many network access devices are expected to implement both the 802.1x authenticator role and the RADIUS client role. However, 802.1x is unable to support the challenge-response mechanisms of RADIUS. Where a port ID is not available, such as in wireless situations, an association ID will be used.

(U) The IETF informational RFC 3580 defines specific mappings and special considerations when using both 802.1x and RADIUS. In particular, it defines how to authorize access to a VLAN by leveraging the tunnel attributes of RADIUS. It also discusses specific known vulnerabilities with RADIUS and EAP and provides approaches to mitigate them.

(U) IETF informational RFC 3579 specifies how a RADIUS client, or a network access server, encapsulates EAP packets to forward to the RADIUS server, where method-specific code can interpret and process the requests. This characteristic enables the network access server to be neutral as to which authentication method is being used and to be unaffected by the introduction of new authentication methods.

2.1.3.3.4.4 (U) PANA

(U) A more recent standards initiative is underway in the IETF work on a Protocol for carrying Authentication for Network Access (PANA). This work is still in a draft status, with additional deliverables planned for 2004 to define the interactions between PANA and 802.1x and to specify a Management Information Base (MIB) for the protocol.

(U) Goals for the PANA effort include support for roaming devices, dynamic choice of service providers, and multiple authentication methods—all based on IP protocols. PANA is designed to work with EAP as a network-layer transport, carrying EAP payloads independently from the choice of link-layer protocol and avoiding potential roundtrip delays during connection establishment. Note, however, that the primary focus of this effort is to authenticate devices at Layer 3 or above before granting use of network services. A typical usage scenario involves a client system authenticating to a server to gain network access.

(U) While mechanisms such as 802.1x and PPP already support specific link-layer support for EAP, other application-layer authentication approaches are considered to be ad hoc and vulnerable.

(U) The work on PANA is still at an early stage and is being driven mostly by vendors, providing wireless network services, and mobile clients.

2.1.3.3.4.5 (U) Platform-Based Key Storage

(U) Hardware key storage is becoming built directly into personal computing devices. The Trusted Computing Group (TCG) and Next Generation Secure Computing Base (NGSCB) allow PKI keys and certificates to be stored on chips, which are manufactured into PC and PDA motherboards. In essence, the personal computing device contains a *built-in* smart card.

(U) Although only a small number of vendors (e.g., IBM and HP) offer such products today TCG and NGSCB will play important roles in digital rights management and platform security in the next few years.

2.1.3.3.4.6 (U) XML and PKI [XKMS]

(U) As mentioned previously, the appeal of XML has reached PKI in the form of XKMS, a lighter-weight approach for clients and servers to deal with some of the complexities of traditional PKI processing, such as certificate path-checking and validation. While XKMS capability is being introduced into newer versions of PKI products, it has not yet had a major impact on the industry.

(U) The World Wide Web Consortium (W3C) has published requirements for Version 2 of XKMS, which intends to improve the XKMS interactions with Simple Object Access Protocol (SOAP), XML Schema, and Web Services Description Language (WSDL).

(U) XML Signature and XML Encryption standards have been formalized by the W3C and promise to be a prevalent part of future application development. The ability to encrypt and sign individual components of XML documents will require robust key management capabilities, a role potentially filled by PKI.

(U) The Organization for the Advancement of Structured Information Standards (OASIS) has initiated a standards process for the XML-based Digital Signature Services (DSS). To date, a draft exists only for requirements and use cases, but DSS intends to provide an overarching set of XML techniques for the processing of digital signatures, including verification, time stamping, and signature creation.

(U) Although ITU X.509 and the IETF PKIX group use ASN.1 as the basis of encoding for PKI certificates, there is interest in creating a general-purpose standard for XML certificate encoding. Discussions in the IETF and W3C have resulted in some initial drafts, but nothing has emerged as a clear standards candidate at this point. Due to the concerns about ASN.1 development and processing complexity, however, it is likely that continued effort in this area will result in the creation of a standards-based XML digital certificate format.

2.1.3.3.4.7 (U) IPsec VPNs

(U) Two headers form the basis of IPsec: the Authentication Header (AH) protocol and the Encapsulating Security Payload (ESP) protocol. AH, as the name implies, is used for authenticating packets from a host or network device. The ESP header can be used for both authentication and encryption.

(U) Each of these protocols can operate in one of two modes: the transport mode or the tunnel mode. In transport mode, the protocol operates primarily on the payload of the original datagram. In tunnel mode, the protocol encapsulates the original datagram in a new datagram, creating a new IP header and treating the original datagram as the data payload.

(U) The design of the AH and ESP headers is modular, which allows different cryptographic algorithms to be used as needed. As new algorithms are developed, such as elliptic curve algorithms and the Advanced Encryption Standard (AES), the parameters for their use can be standardized within IPsec's architecture and then used in conjunction with AH or ESP.

(U) Although the AH and ESP protocols do not specify a particular automated encryption key-management system, IPsec implementations are designed to support both preshared keys and the automated key management system called Internet Key Exchange (IKE), which is defined in IEEE RFC 2401.

2.1.3.3.4.8 (U) SSL VPNs

(U) Using SSL version 3.0 to implement secure network connections is different than using IPsec, because connections focus on individual users and sessions rather than on multiplexed communications between sites. Thus, SSL-secured networks are similar to remote access VPNs, although most implementations of SSL-secured networks connect a user to a server (or server farm) and not to all the resources at a site.

(U) One of the most appealing features of using SSL for a secure network is the deployment simplicity. The minimum requirements for an SSL-secured network are a Web server with an appropriate digital certificate and a Web browser on each user's computer. Note that this setup is mostly used for Web-based access. File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP), and Network News Transport Protocol (NNTP) can use SSL if the appropriate SSL-enabled versions of those products are used.

(U) As commonly deployed, only the servers require digital certificates to initiate SSL sessions. This considerably reduces the number of certificates to be managed and distributed. That may suit some enterprises. However, organizations looking to authenticate external users, such as for an extranet, must employ some form of client authentication. This adds the requirement for a PKI system if authentication is to be performed within the SSL protocol.

2.1.3.3.5 (U) Costs/Limitations

(U) Device authentication technologies and protocols, while existing in some form today, are still considered emerging technologies. This can be seen in the Standards section, while noting the number of Working Groups (IETF and others) that are still working towards enhancing the authentication and security of these standards.

(U) From a pragmatic GIG enterprise services viewpoint, the type technology selected depends on the particular situation and its mission needs of the authentication strength. For example, for situations that do not require the strictest authentication and secure levels, combinations of Wi-Fi Protected Access (WPA) on a wireless local area network (WLAN) using RADIUS and LDAP servers should meet most needs.

2.1.3.3.6 (U) Dependencies

(U) Microsoft provides built-in support for 802.1x in Windows XP. Windows 2000 users running Service Pack 3 can download the Microsoft 802.1x Authentication Client for Windows 2000. Microsoft also supplies versions of this client software to Windows 98 and NT users with a Premier support agreement.

(U) Apple has built-in support for 802.1x in Mac OS X (v10.3), which can be configured to access either an AirPort wireless connection or a secure Ethernet port. Mac OS X v10.3 also supports WPA for WLANs without the need for 802.1x or a RADIUS server, which is ideal for home users without a RADIUS server.

(U) Linux systems require client software that performs the 802.1x supplicant function. This software is available from the Open Source Implementation of 802.1x site used in combination with OpenSSL (Secure Sockets Layer) and FreeRADIUS.

(U) In addition, developer kits and 802.1x drivers for various operating environments are available from software vendors, such as Meetinghouse Data Communications with its AEGIS product line. The AEGIS client is available for Windows 98, ME, NT, 2000, and XP; Pocket PC and Palm products; Mac OS X; and Linux. Funk Software offers its Odyssey client for Windows 98, ME, NT, 200, and XP; Pocket PC; and Windows Mobile.

(U) A growing class of products that assess the status of client systems for conformance to security policies are embracing 802.1x authentication to integrate with network switching systems. Access to the network is only granted once policy conformance has been established. Both Zone Labs' Integrity 5.0 and Sygate's Secure Enterprise support this feature. Zone Labs (acquired by Checkpoint Software) certifies its 802.1x feature to work with products from Aruba, Cisco, Enterasys, Funk Software, and Microsoft. Sygate announced support for interoperability with products from Cisco, Enterasys, Extreme, HP, and Nortel. One of the features of Sygate's solution is to quarantine any client systems, which are not running policy-checking agent software, to a guest VLAN.

(U) Other third-party software products inherit support for 802.1x simply by working with existing 802.1x-aware client software, such as the support built in to Windows XP. For example, RSA provides support for SecurID authentication to WLANs through its Advanced Computing Environment (ACE)/Agent for Windows and the Windows XP wireless LAN client.

(U) Fiberlink, GRIC, and iPass are implementing similar capabilities for their VPN clients. These companies provide remote access management and VPN capabilities. Their clients check the mobile device infrastructure to make sure—before allowing connection—that the firewall is running, the virus scanner is running and up to date, and the VPN is active.

2.1.3.3.7 (U) Alternatives

2.1.3.3.7.1 (U) MAC/IP address

(U) An alternative is to use the older simpler methods of device identification such as the media access control (MAC) address or IP address of the device the user is using at the time. Enterasys' User Personalized Network (UPN) is such an example. Once identity is established, the switch can determine whether to grant access to devices associated with a restricted VLAN. One of the main strengths of the UPN is its ability to provision network services and applications based on user identity. The Enterasys solution relies on existing enterprise investments in directories—such as Microsoft's Active Directory or Novell's eDirectory—to authenticate user identity and establish an association with the user's location.

(U) Within the scope of device authentication, there exist a number of alternatives and combinations. Most of these are related to specific vendors and platforms. These are described below.

- (U) Alcatel implements an approach to Layer 2 authentication within its OmniSwitch product line. Alcatel's authenticated VLAN (AVLAN) feature does not rely on operating system support for EAP and 802.1x, but requires an Alcatel-supplied client application: AV-Client for Windows 9x, NT, 2000, and XP. This client combines the Windows login with a network login, so a user enters an identity and credential only once. A successful authentication connects the user to the VLAN and its resources.

- (U) Cisco has a framework for identity-based networking services that is supported across several product lines, including Catalyst switches (6500, 4500, 3550, and 2950), Aironet wireless access points, and Cisco's Secure Access Control Server v3.2 (ACS). The various network switch products implement 802.1x. They perform the role of an authenticator or intermediary between the supplicant at the client and the RADIUS authentication service. Cisco's RADIUS server product is ACS.

- (U) Cisco extends 802.1x to enable dynamic assignment of VLANs to ports (based on identity), guest VLAN support, mapping of access control lists (ACLs) to a port based on the user's 802.1x identity, and synchronization of port security status in case of failover. Also, Cisco IP phones can be automatically mapped to a voice VLAN when detected. Computers connected to IP phones will need to authenticate to get access to the network.

- (U) Cisco also announced its Network Admission Control (NAC) program, a collaboration with industry partners focused on limiting damage from security threats originating at client systems that have been compromised by a virus or worm. In its initial phase, NAC enables Cisco routers to enforce access privileges when an endpoint device attempts to connect to a network. This decision can be based on information about the endpoint device, such as its current antivirus state and operating system patch level. NAC allows noncompliant devices to be denied access, placed in a quarantined area, or given restricted access to computing resources.

- (U) Nortel has supported 802.1x in its BayStack switches since 2001. Recent extensions to its BayStack operating system Switching Software (BoSS) v3.0 for BayStack 420 and

425 switches, improve its support for EAP and 802.1x. Access to network services requires a login to a RADIUS authentication server. Also, its Wireless LAN 2200 series includes support for Virtual Port-based Authentication (VPA) based on EAP and 802.1x back to a RADIUS server (both in its WLAN Access Points and in the optional WLAN Security Switch 2250 unit). Other products, such as Passport 8600, support VLANs for a variety of network separation requirements. Nortel partners with Sygate to leverage 802.1x to quarantine systems that are out of compliance with local security configuration policies.

2.1.3.3.7.2 (U) VPN-based Authentication

(U) IPsec-based VPN: Due to its original development for site-to-site VPNs, IPsec focuses on machine authentication rather than user authentication, and this has caused problems in creating interoperable dial-in clients. To improve the usability and interoperability of IPsec-based VPN dial-in clients, the IETF's IPsec Remote Access (IPSRA) working group has been trying to settle on a single protocol that it will propose as a standard to the IETF. After almost two years' work on four (or more) different proposals, the working group has settled on the Pre-IKE Credential Provisioning Protocol, or PIC, which is slowly making its way into commercial products.

(U) SSL-based VPN: Though the SSL standard does not support client authentication methods other than digital certificates, it is possible to use other authentication methods in conjunction with SSL. The simplest approach is username and password, but it is also possible to use stronger authentication methods, such as security tokens or smart cards.

2.1.3.3.8 (U) References

(U) An Initial Security Analysis of the IEEE 802.1x Protocol -
(U) http://www.cs.umd.edu/~waa/1x.pdf.

(U) Ross Anderson's Home Page - http://www.cl.cam.ac.uk/users/rja14/#Reliability.

2.1.3.4 (U) Authentication Protocols

2.1.3.4.1 (U) Technical Detail

(U) There are two major traditional authentication protocol techniques – Symmetric Key Authentication and Public Key Authentication.

(U) Symmetric Key Authentication:

(U) In symmetric key authentication, the shared secret key is used at the client to create an OTP that is then transmitted to the server. The same process is done at the server, and if a match exists, the user is authenticated.

(U) Many commercial schemes use public-domain hash functions based upon ANSI X9.9, which relies on Data Encryption Standard Message Authentication Code (DES MAC), which is a cipher-block, chained checksum. Some vendors use proprietary algorithms, such as RSA Security. It should be noted that X9.9 (based on 56-bit single DES) was withdrawn by ANSI in 1999 in favor of the stronger Triple DES algorithm.

(U) Another often used public domain hash function is the SHA-1 or Secure Hash Algorithm, which comes from NIST in the federal government. For greater security, some tokens actually recalculate a new-shared secret key after each authentication process, which requires that the server do likewise in order to keep in step.

(U) A common symmetric key authentication scheme is the Kerberos protocol. Kerberos is a network authentication protocol. Kerberos is designed to provide strong authentication for client/server applications by using secret-key cryptography. This is accomplished without relying on authentication by the host operating system, without basing trust on host addresses, without requiring physical security of all the hosts on the network, and under the assumption that packets traveling along the network can be read, modified, and inserted at will. Kerberos performs authentication under these conditions as a trusted third-party authentication service by using conventional cryptography, i.e., shared secret key. The authentication process proceeds as follows:

1. (U) A client sends a request to the authentication server (AS) requesting "credentials" for a given server.
2. (U) The AS responds with these credentials, encrypted in the client's key. The credentials consist of 1) a "ticket" for the server and 2) a temporary encryption key (often called a "session key").
3. (U) The client transmits the ticket (which contains the client's identity and a copy of the session key, all encrypted in the server's key) to the server.
4. (U) The session key (now shared by the client and server) is used to authenticate the client, and may optionally be used to authenticate the server. It may also be used to encrypt further communication between the two parties or to exchange a separate sub-session key to be used to encrypt further communication.

(U) Another symmetric key authentication protocol is CHAP or the Challenge Handshake Authentication Protocol (defined in RFC 1994) verifies the identity of the peer using a three-way handshake. The following general steps are performed in CHAP.

1. (U) After the link esStablishment phase is complete, the authenticator sends a challenge message to the peer.

2. (U) The peer responds with a value calculated using a one-way hash function (Message Digest 5 [MD5]).

3. (U) The authenticator checks the response against its own calculation of the expected hash value. If the values match, the authentication is successful. Otherwise, the connection is terminated.

(U) Public Key Authentication:

(U) Unlike symmetric key authentication which relies on a single shared secret key, public key authentication employs a related pair of keys: one public (known to the server) and one private (known only to the client token and computationally unlikely to be derived from its public key counterpart). In the authentication process, the token employs its private key in a cryptographic function related to that which is executed by the server with the public key. The token function is typically implemented as a software token on the local client host, usually in a challenge-response mode.

(U) Effective management of public and private key pairs across a large population of users requires a PKI. A public key certificate (or digital certificate) binds a user identity with its associated public key, and a trusted central agent or certification authority (CA) serves to verify the validity of issued certificates.

(U) In a challenge-response authentication process, the server would send a random challenge to the client. The client then uses its private key to digitally sign the challenge, which is then returned as a response to the server along with its public key certificate (which could alternatively be retrieved by the server from the CA). If the certificate is shown to be valid, the server verifies the digital signature through application of the client's public key.

(U) Currently deployed examples of public key certificate-based software token authentication include Microsoft's Windows 2000 server operating system (using PKINIT or Public Key Initialization Authentication) and commercial versions of Secure Shell (SSH).

(U) Authentication mechanisms often depend upon the environments in which they are to operate, along with other considerations. The following sections describe various aspects of emerging authentication technology.

2.1.3.4.1.1 (U) 802.1x for network applications

(U) For network access applications, 802.1x can serve as the authentication protocol framework. This is true both for wired and wireless networks The authenticator is the access point for wireless networks; it is the layer-two switch for wired networks. Figure 2.1-5 shows a network authentication framework. A natural candidate is 802.1x because it already defines EAP methods for each of the proposed base authentication methods (e.g., EAP-SIM for SIM-based authentication, EAP-TLS for PKI-based authentication, and EAP-PEAP for OTP-based authentication).

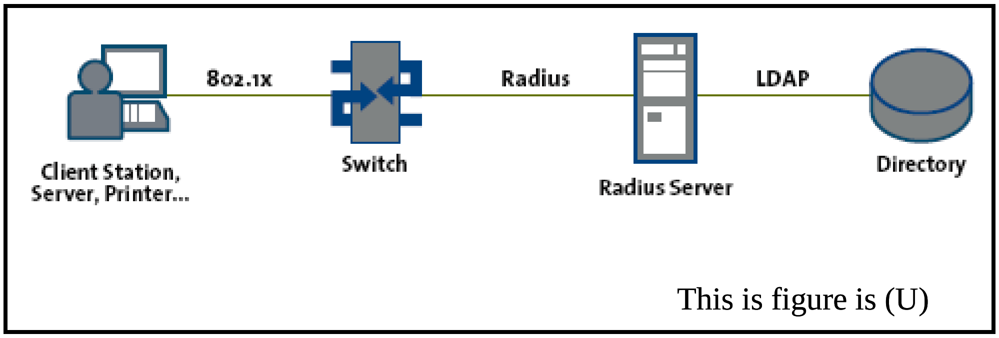


Figure 2.1-5: (U) Network Authentication Framework

2.1.3.4.1.2 (U) 802.1x for device authentication

(U) The 802.1x framework is crucial to promote a consistent deployment profile for device authentication across manufacturers and OS vendors. Embedded 802.1x clients can be deployed to enable these devices (e.g., VoIP phones, access points, switches, servers) to transparently authenticate to the network, before being handed an IP address and being granted access to the network. Figure 2.1-6 shows this.

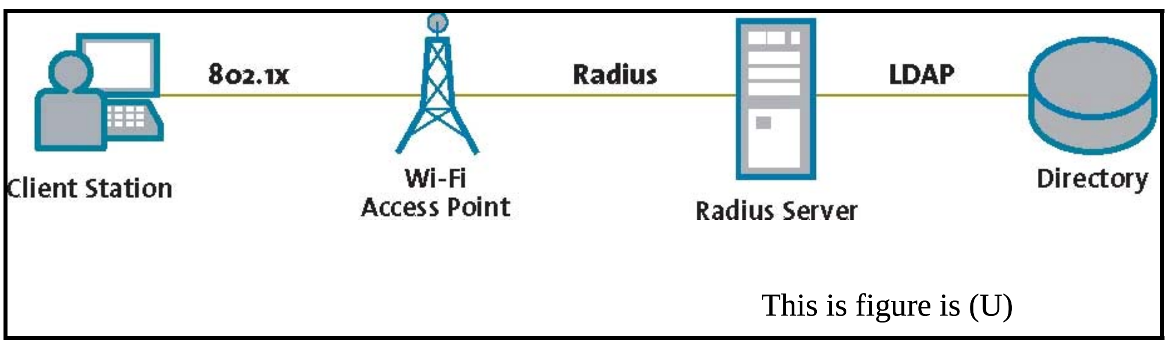


Figure 2.1-6: (U) Device Authentication Framework

2.1.3.4.1.3 (U) Manufacturing-time device credentials

(U) Device certificates can be combined with emerging secure computing technologies such as the Trusted Platform Module (TPM) and the 802.1x authentication protocol framework. This convergence will foster a common technology stack and deployment profile to allow device manufacturers to enable turnkey-strong device authentication solutions. In fact, using the established profile, manufacturers and OEMs will be able to rapidly collaborate to embed the necessary hardware credentials and client software at manufacturing time.

2.1.3.4.1.4 (U) Web service protocol for business-application integration

(U) Universal strong authentication must address the protocol dichotomy between network access applications (e.g., dial-up, VPN, Wi-Fi) and business applications, such as Web or enterprise portals, Web applications, ERP systems, and Web services. The 802.1x framework is particularly well suited to the former, but not to the latter. A Web service interface is better adapted to today's business applications.

(U) Because the authentication protocols constitute the primary mechanism for integration into applications, open authentication requires a palette of protocols that can support both types of applications. This requirement leads to the definition of a Web service API alongside the 802.1x EAP methods already covered. The Simple Object Access Protocol (SOAP) API can leverage the WS-Security specification as the primary mechanism for encoding the base security tokens (OTP, X509 certificate). It can also define a challenge-response mechanism for SIM-based authentication.

2.1.3.4.1.5 (U) Application connectors and authentication clients

(U) The main motivation for standardizing an authentication protocol and promoting the development of authentication clients is to foster the creation of application *connectors*. Application connectors, or agents, are the client libraries of strong authentication. They must be portable across major operating systems and offer APIs across popular languages. Such flexibility would make it easier for application developers to integrate strong authentication within custom applications (e.g., link, compile, and run). This is mainly true for the EAP protocols—EAP-SIM, EAP-TLS, EAP-PEAP—because the Web service can immediately leverage the Web services stack that exists in all major development platforms.

2.1.3.4.1.6 (U) Credential Provisioning and Validation

(U) Since universal strong authentication is a key objective, the blueprint needs a method to harmonize credential issuance and other life cycle management functions across all types of secrets, symmetric keys or RSA key pairs. The SIM and OTP secrets become subordinate to an RSA key pair (a device certificate key pair). The shared secrets are encrypted and embedded as attributes within the certificate. The certificate acts as a private store for the shared secrets, and the security device acts as a secure hardware vault for the *root* credential.

(U) This approach allows manufacturers and customers to leverage the breadth of secret management capabilities and security practices (e.g., key escrow, secure roaming, and directory services) from existing PKI platforms. The method applies both to secure device personalization (shared secret and device certificates embedded at manufacturing time) and secure provisioning of user credentials. This unified credential life cycle management framework will leverage existing public key cryptography standards and modern protocols such as XML Key Management Specification (XKMS).

(U) Validation profiles will be defined by the choice of authentication protocols, as described earlier. In addition, validation services will be able to validate X.509 certificates using certificate revocation lists (CRLs) and industry standards such as Online Certificate Status Protocol (OCSP) or XKMS.

(U) Validation servers in a strong authentication environment have the same characteristics as RADIUS servers. This is a conscious choice, as RADIUS servers are already a key component of an ISP or enterprise network infrastructure. Furthermore, high-quality RADIUS servers are widely available from vendors and open-source projects. The complexity and cost overhead for deploying strong authentication can be reduced by leveraging the large, existing, installed base of RADIUS servers.

(U) For applications that require a Web service interface, the validation server will be required to implement the SOAP validation protocol discussed earlier. In the network world, the strong authentication validation server is congruent to a RADIUS server; while in a service-oriented architecture, the validation server is an instance of a Web service. Because credential validation is highly complementary to credential mapping and exchange, it makes sense to consolidate Web services with the architectural concept of Security Token Service (STS), as defined by Web Services Trust Language (WS-Trust).

(U) An important architecture goal for universal authentication is to enforce the separation between validation and identity stores. All identities (user or device identities, as well as device-to-user bindings) should be maintained outside the validation server. This separation is important from an integration and cost-control standpoint. It promotes a distributed architecture that favors the reuse of an enterprise's existing infrastructure (e.g., corporate directories). In such an architecture, the validation server is a minimal front end.

2.1.3.4.2 (U) Usage Considerations

(U) In many cases, the specific application dictates the authentication protocol. For example, in a Web application, TLS will often be the primary protocol. In the VPN case, IPsec IKE is the standard, and for wireless Wi-Fi (802.1x), Extensible Authentication Protocol (EAP) methods such as EAP-TLS or EAP-PEAP are the norm.

(U/FOUO) A major disadvantage of symmetric key authentication is that it does not scale well to large and global user populations, due to the logistical difficulties of distributing the shared secret keys. This disadvantage affects the use of the following protocols:

- (U) Kerberos

- (U) CHAP
- (U) 802.11 wireless
- (U) EAP-PEAP for OTP (one-time-password) authentication

2.1.3.4.2.1 (U) Advantages

(U/FOUO) A distinct advantage of public-key authentication is that it easily scales to very large networks (such as the GIG), whereas symmetric key or shared-secret authentication is generally limited to specific communities of interest in which the key management process will not be unduly burdensome.

2.1.3.4.2.2 (U) Risks/Threats/Attacks

(U/FOUO) A common risk/threat/attack that has to be anticipated and dealt with appropriately by any proposed authentication scheme is the classic man in the middle (MITM) attack in which a malicious adversary will intercept the communications between a client and its authentication server, and then modify the message protocol contents so as to defeat, hijack, or otherwise maliciously alter the proper authentication protocol. It is essential that all critical authentication messaging be suitably encrypted so as to prevent this.

2.1.3.4.3 (U) Maturity

(U) Due to the strong desire across both the government and industry (particularly the financial industry) for secure authentication of parties conducting electronic communications and transactions, authentication protocols have developed over the years into a fairly mature state. Thus, the Technology Readiness Level of authentication protocols would be grouped into the Mature category (TRL 7 – 9).

2.1.3.4.4 (U) Standards

(U) There are a variety of formalized international and American standards covering the technology of authentication protocols.

2.1.3.4.4.1 (U) International Standards:

(U) The international standards bodies that are responsible for developing authentication protocols include:

- (U) IETF Internet Engineering Task Force (http://www.ietf.org)
- (U) ISO International Organization for Standardization (http://www.iso.ch)
- (U) ITU-T International Telecommunication Union Telecommunication Standardization Sector (http://www.itu.int/ITU-T)
- (U) IEEE Institute of Electrical and Electronics Engineers (http://grouper.ieee.org/groups/1363/)
- (U) Industrial consortiums such as OASIS (Organization for the Advancement of Structured Information Standards, http://www.oasis-open.org/committees/wss), which develops

security standards for web services

(U) IETF standards that are relevant to authentication tokens include Internet Drafts from the Secure Shell working group, and RFCs 2289 and 1760 that describe the S/Key One-Time-Password System.

(U) Relevant ISO standards include ISO 8731 (algorithms for banking message authentication), ISO/IEC 9797 (MACs via block cipher and hash function), ISO/IEC 9798 (entity authentication by symmetric, digital signature, and cryptographic check), and ISO/IEC 19092.

(U) Relevant ITU-T standards include those describing directory certificates for authentication such as X.509 (issued 08/97, authentication framework), and X.509 (issued 03/00, public key and attribute certificate frameworks).

(U) IEEE standards include P1363 (specifications for public key cryptography).

(U) OASIS standards include WSS (Web Services Security) Version 1.0 (April 2004). WSS handles confidentiality/integrity for SOAP (Simple Object Access Protocol) messages, providing a mechanism for associating security tokens with message content. WSS is extensible and supports multiple security token formats. It builds upon existing security technologies such as Extensible Markup Language (XML) Digital Signature, XML Encryption, and X.509 Certificates to deliver a standard for securing Web Services message exchanges. Providing a framework where authentication and authorization take place, WSS lets users apply existing security technology in a Web Services environment.

(U) Founded in 1993, OASIS has members in 100 countries and 600+ organizations (including Entrust, HP, Hitachi, IBM, Microsoft, Nokia, RSA Security, Sun Microsystems, and Verisign).

2.1.3.4.4.2 (U) American Standards:

(U) Organizations in the United States that are responsible for developing and promulgating authentication protocol standards include ANSI American National Standards Institute (http://www.ansi.org), and NIST National Institute of Standards and Technology (http://www.itl.nist.gov/fipspubs, repository of the Federal Information Processing Standards or FIPS).

(U) Relevant ANSI standards include X9.9 (message authentication codes for symmetric token authentication, withdrawn in 1999 due to attacks demonstrated against single DES 56-bit key, in favor of double or triple DES), X9.30 (public key cryptography, digital signature algorithm DSA, secure hash algorithm SHA-1, DSA certificate management), X9.31 (reversible public key cryptography for digital signatures rDSA), X9.45 (management controls using digital signatures and attribute certificates), X9.52 (triple DES modes of operations), X9.63 (key agreement and transport using elliptic curve cryptography ECC), X9.71 (keyed hash for message authentication), and X9.72 (peer entity authentication using public keys).

(U) Relevant NIST FIPS PUB standards include FIPS 180 (secure hash algorithm SHA-1), FIPS 186-2 (digital signature standard DSS, same as ANSI X9.30), FIPS 190 (guideline for use of advanced authentication technology alternatives), FIPS 196 (entity authentication using public key cryptography, same as ANSI X9.72), and FIPS 197 (advanced encryption standard AES). An informative new NIST draft document on authentication mechanisms is Special Publication 800-63 (*Recommendation for Electronic Authentication*, January 2004, which can be found at http://csrc.nist.gov/publications/drafts/draft-sp800-63.pdf).

(U) The purpose of this section is not to explain all of the various algorithms used by authentication tokens but to note that tokens—hardware or software—can use a variety of cryptographic algorithms to produce the desired OTP (algorithms such as DES, Triple-DES, DSA, SHA, ECC, and the new AES Advanced Encryption Standard). However, as algorithms are improved and attacks discovered against the weaker algorithms, some standards are superceded or withdrawn.

2.1.3.4.5 (U) Cost/Limitations

(U) An authentication protocol that is based upon symmetric or secret key cryptography has in it a very costly and limiting characteristic in that the associated secret keys must be delivered a priori to all parties. This is a severe limitation in the context of the GIG.

(U) Whereas both symmetric and public key authentication can be done at the application layer, only public key authentication can be done automatically at the transport layer.

2.1.3.4.6 (U) Dependencies

(U) One dependency of public key encryption-based authentication protocols is the existence of a well-developed and robust PKI.

2.1.3.4.7 (U) Alternatives

(U) The alternatives to use of an authentication protocol are few and undesirable. One alternative is simply to forgo authentication, but this is not thinkable in the context of the GIG. Another alternative would be within the context of a closed system where all communicating/participating parties are talking securely to each other over link-encrypted lines and are thus inherently trusted to each other.

2.1.3.4.8 (U) Complementary Techniques

(U) Certainly tokens (both hardware and software) are a complementary technology to that of authentication protocols. It is within the client-retained token that much of the authentication algorithm is either stored and/or executed in the field during a given authentication attempt.

2.1.3.4.9 (U) References

(U) RFC 1994, “PPP Challenge Handshake Authentication Protocol (CHAP)”, http://www.ietf.org/rfc/rfc1994.txt , by W. Simpson, 1996.

(U) NIST Special Publication 800-63, “Recommendation for Electronic Authentication”, http://csrc.nist.gov/publications/drafts/draft-sp800-63.pdf, January 2004.

2.1.3.5 (U) Authentication Confidence

(U) Authentication confidence refers to developing a system that determines the probability that a user or other device is who he/she/it claims to be. It takes into account such factors as:

- (U) The authentication mechanism (e.g., static password, public-key cryptography, software token, hardware token, biometrics)
- (U) The authentication protocol used: e.g., a protocol that is known to be secure against man-in-the-middle attacks or one that is based on strong cryptographic operations
- (U) The location of the entity being authenticated: e.g., a secure office, CONUS or OCONUS, a public kiosk or Internet cafe, a tactical battlefield
- (U) Characteristics of the device used to authenticate: e.g., a COTS computer owned and controlled by the US Government; a publicly-accessible COTS computer; a dedicated, tamper-resistant device
- (U) The communications path between the entity being authenticated, and the server providing authentication and/or access decisions: e.g., a secure, U.S. Government-owned or leased network; a wireless network on a battlefield; commercially-provided telecommunications lines; a coalition partner's network

(U//FOUO) The goal of authentication confidence is to quantify the risk that a user or entity attempting to access the system is not the purported user or entity. This risk can then be provided to an access control service to grant or restrict access to system resources.

(U//FOUO) The simplest example of authentication confidence is a user logging into the system over an insecure network, from a public kiosk, using a static password based authentication system. For example, someone purporting to be Joe logs into the system and provides the correct password. However, from tracing IP addresses and using known information, the authentication server determines that Joe is coming in over a public Internet Service Provider's network from a public kiosk in a coffee shop and is not using a strong authentication protocol. How confident is the authentication server that this is really Joe, when there are numerous opportunities for the password to have been compromised? It could have been acquired previously through a dictionary attack or by someone finding a slip of paper with Joe's password. It could have been captured on this use, via a keystroke logging function on the public terminal, or at some point over the network. Thus, even though some entity has provided a valid user identifier and the correct password, the system may still want to limit or even prevent access to resources, for fear that the entity at the other end of the connection is not really Joe. This may be the case for future login sessions as well, as Joe's password now is very likely to have been compromised upon this use.

(U//FOUO) Note that authentication confidence is related to but distinct from policy-based access control decisions. In the scenario described in the previous paragraph, the result of a weak level of confidence in Joe's authentication was that Joe was restricted from or prevented from accessing certain resources. This is because authentication confidence is one of a number of inputs to the access control mechanism. However, other inputs to that mechanism could have also resulted in access being restricted. For example, even if there was perfect confidence that Joe was really the user accessing the system, and that there was no chance that Joe's authentication data was compromised for future uses, Joe's access might still be restricted because of his location or communications path (e.g., sensitive or classified information would not be sent to a location with insufficient physical security).

2.1.3.5.1 (U) Technical Detail

(U) Authentication confidence at this time is a research area. While some work has been done, and the general requirement is understood, there are significant details to be worked out and major questions to be resolved. Among the issues to be addressed are:

- (U) Authentication metrics: It is generally accepted that static passwords are weaker than one-time passwords, and that a hardware token with a PIN is generally better than a software token. However, there is no quantitative metric that compares different types of biometric authentication with each other or that compares biometric authentication with hardware token-based authentication or public-key cryptography-based authentication. In order for authentication confidence to have any meaning, there must be a way to measure and determine the relative (if not absolute) strength of each given authentication method.

- (U) Reliable communication of user location: One of the factors normally considered to be part of authentication confidence is the location of the user, e.g., within a secure area or in public. In order for authentication confidence to be used, there must be a way for the authentication server to reliably know this information. The information must be conveyed to the server, and it must not be possible for an attacker to spoof this. For example, it must not be possible for a public terminal in a kiosk to convince the authentication server that it is in a secure location; and it must not be possible for a device that is on a battlefield in Southwest Asia to convince an authentication server that it is in a headquarters building in CONUS.

- (U) Reliable communication of device characteristics: Another factor of authentication confidence is the characteristics of the device being used by the user (e.g., a public COTS computer system, a COTS computer system controlled by the Government organization, or a special-purpose device with strong tamper resistance and strong cryptography). The device must be capable of communicating this information to the authentication server, and it must not be capable of being spoofed. One of the initial research areas is determining precisely which set of characteristics is important in which situations.

- (U) Corrections/modifications for error cases: For every type of authentication system used, there are two possible types of errors: false positives, in which the wrong entity is authenticated as being the correct one; and false negatives, in which the correct entity is rejected. Each authentication technique has different false positives and false negatives. For a password-based system, a false positive occurs when an attacker knows the correct password; a false negative occurs when the legitimate user fails to enter the correct password (because he has forgotten it or mistyped it). For a biometric-based system, false positives occur when an attacker's measurement is close enough to the legitimate value to allow authentication. For a false negative to occur the legitimate user's value is rejected as not matching the stored value. In traditional authentication systems, these differences can be taken into account by policy, but the bottom line is that a user is authenticated or not as a binary state. A user who is deemed to match gets access; one who is deemed not to match is rejected. There is no partial authentication or reflection of potential errors. One of the potential benefits of an authentication confidence system is that it allows for partial access, based on a partial match. That is, the authentication server could decide that a fingerprint is close enough to the correct value to allow some access, but there is enough doubt (i.e., through possibly smudged lenses, scraped-off fingerprints) that access to the most sensitive information and resources will be withheld. This results in allowing legitimate users some use of the system so that they are not completely shut out, while restricting the amount of damage that an attacker can cause.

2.1.3.5.2 (U) Maturity

(U) As this is a research area at the present time, there are no significant usage considerations to document. As the area matures, usage will be a major factor in the development and deployment of authentication confidence mechanisms and solutions.

(U) At this point, authentication confidence is in its infancy, and thus is assigned to the lowest Technology Readiness Group: Early (TRL 1 – 3).

2.1.3.5.3 (U) Standards

(U) A major step necessary for acceptance of authentication confidence metrics will be standards for those metrics. Without standards, users and organizations will not be able to assign meaningful values and make appropriate decisions about allowing access. In particular, standards will need to address:

- (U) Authentication metrics. In addition to standards for the individual authentication mechanisms (e.g., passwords, biometrics, and authentication tokens), standards will be needed to map the metrics to one another

- (U) Error indications: Standards will be required for assessing "how close" a presented authenticator is to the "correct" one; e.g., a biometric value was deemed to be incorrect, but it was off by some small value; or a password presented was not the correct one, but it differed from the correct one by some characteristic which could easily be explained by a typing error or line noise.

2.1.3.6 (U) Single Sign-On

(U) Single Sign-On (SSO) has traditionally been limited to cases covering the one-time sign-on process for access to all services of a single organization, whereas Global Sign-On has applied to multiple participating organizations that had reached an a priori collaborative agreement to avail users with a common sign-on process. In the GIG Vision SSO is expanded to enable a user to login or sign-on only once to a global authentication server thus allowing an entity to simultaneously access the GIG information and resources without any requirement for additional identification and authentication. With this definition, SSO and Global Sign-On become one and the same. Some communities view Global Sign-on as including the issues related to mobile users, while SSO does not. In this document fixed versus mobile issues are both treated under SSO.

(U) The goal of an ideal SSO system is to enable a user to login or sign-on only once to a global authentication server. This approach eliminates the need to enter different passwords to login to a workstation, to each service, database, etc. and replaces this with an automatic sign-on or re-authentication of an entity, making sign-on transparent. SSO must not sign an entity on with all of their privileges or escalate an entity's privileges without the entity's consent. This would be equivalent to signing on as a system administrator/super user to read personal email. SSO should also include a way to lower (or release) privileges once the activity that required increased privileges is complete.

(U/FOUO) The initial sign-on process must be very robust and secure and based upon the ancillary enabling technologies of biometrics, multi-factor authentication, tokens, one-time passwords, and/or strong session establishment protocols. Once the server is certain as to the entity's identity, that entity's global credentials and/or roles would be provided back to the entity (e.g., as a ticket, certificate, or SAML assertion), thus enabling follow-on transparent login to all network resources and applications that are allowed.

(U) Since the credentials/roles are critical, if and when they are sent to the local user client end, they should be managed and processed only by trusted hardware (e.g., a hardware token or smart card) that would be immune to malicious sniffing, viruses, or Trojan horses. Transmission of credential information should be done encrypted so as to protect it while it is in transit.

(U/FOUO) All of the above merely emphasize that SSO technology has the unavoidable effect of concentrating much potential, aggregated risk in a small number of processes and information repositories. Nevertheless, the convenience and utility of SSO to the average user is such that the GIG is certain to feature SSO capabilities. As such, a successful SSO architecture fruition within the context of the GIG will demand very strong and mature identification and authentication technologies at the front end along with a robust privilege management infrastructure at the back end.

2.1.3.6.1 (U) Technical Detail

(U) SSO capabilities have been evolving over a number of years in commercial applications. SSO has been enabled by a number of technical advances, including strong authentication techniques, biometrics, and tokens (which allow one-time passwords).

2.1.3.6.1.1 (U) Early SSO Techniques

(U) A number of methods have been used over the years by organizations in order to implement techniques that in limited ways approximate the functionality of SSO. These include login scripting, password synchronization, and Lightweight Directory Access Protocol (LDAP) directories, as described below.

2.1.3.6.1.1.1 (U) Scripting

(U) Initial commercial techniques developed for SSO included scripting, whose primary goal is the simple automation of the login procedure, rather than the security enhancement of application access. In scripting, a user conducts a primary authentication to a SSO authentication server. In subsequent accesses to various target systems, the client intercepts the standard login dialogue and then retrieves the appropriate login script from a repository. The client software then merely forwards the *credentials* (which may merely be an instance of a user ID and password) to the target system via the login dialogue, achieving a transparent automation of the login procedure on behalf of the user. The login script repository may reside within the SSO server or may be downloaded to the client and cached locally.

2.1.3.6.1.1.2 (U) Password Synchronization

(U) As can be seen from the above description, scripting is merely a forced automation of the login procedure across various target systems—each of which may have unique User IDs and/or passwords associated with a specific user. An evolution of this technique is the concept of Password Synchronization, in which a password is shared across various systems and can be updated in a synchronous fashion across all the target systems.

(U) Automatic password synchronization ensures that when a user modifies the password, that new password is routed network-wide to other target systems. Applying password synchronization and self-service password reset technologies reduces the number of unique passwords that a user needs to remember. However, while password policies could be strengthened for passwords that would be reused to access multiple applications and resources (with resulting risk aggregation), there is often still a need for the user to respond to each application's unique login prompt.

2.1.3.6.1.1.3 (U) LDAP directories

(U) Other technologies have also contributed to reducing the number of unique sign-ons that are needed. Fewer application-specific login prompts are required as applications are upgraded to new software that offers integrated support for authentication to a shared Lightweight Directory Access Protocol (LDAP) directory. LDAP directory-based authentication generally involves storing only the cryptographic hash of the user's password, and it may not provide the contextual credential information about password policies and expiration dates.

(U) Each application would require its own logic to support authentication based on the LDAP and the credentials maintained in the directory. Through the enabling of LDAP authentication for target systems, user password information could be made retrievable from any LDAP-supporting network directory. Each user then has only one password—the LDAP password— to gain access to all LDAP-enabled target systems.

(U) LDAP authentication employs the Simple Authentication and Security Layer (SASL) protocol implemented between client systems and the directory server. IETF RFCs, which discuss SASL, include RFC 2222 (*Simple Authentication and Security Layer*, http://www.ietf.org/rfc/rfc2222.txt, by J. Myers, 1997) and RFC 2244 (*The One-Time Password SASL Mechanism*, by C. Newman, 1998). In reality, LDAP authentication only provides for consolidated sign-on rather than true SSO. The user must authenticate separately on each target system. Functionality and benefits similar to password synchronization are provided by LDAP authentication. A potential limitation is that each possible target system must support the LDAP protocol. Nevertheless, LDAP can still effectively reduce the complexity of password management within an enterprise.

(U) The advent of strong multi-factor authentication techniques (leveraged upon the enabling technologies of biometrics, tokens, and one-time passwords) has made it possible to evolve more fully integrated SSO systems that rely upon the initial very robust authentication to an authentication server. Then, the as-needed propagation of (encrypted) authorizing credentials and one-time passwords is sent to each target system as it is encountered. This can follow either a centralized or a federated architecture model.

2.1.3.6.1.2 (U) SSO Architectures

2.1.3.6.1.2.1 (U) Centralized Model

(U) A totally centralized architecture for SSO implementation (as exemplified by the original Microsoft Passport system) is shown in Figure 2.1-7 below.

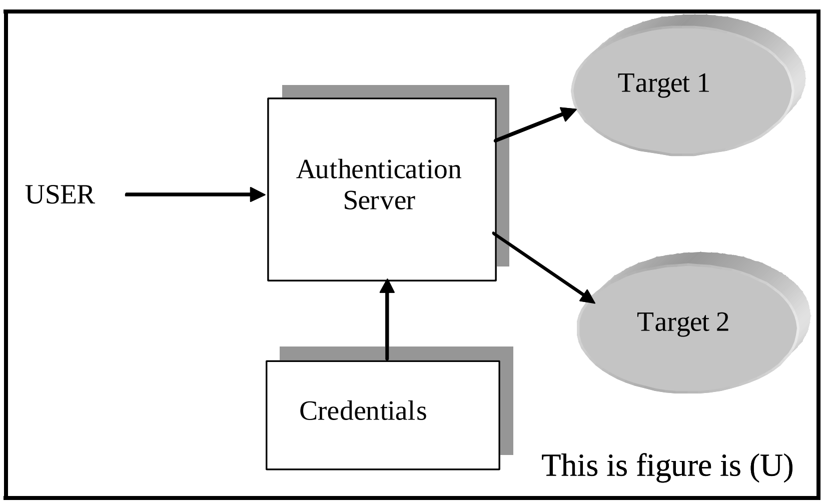


Figure 2.1-7: (U) Centralized Architecture for Single Sign-On

(U) In the centralized model, the user signs on to the centralized gate-keeping authentication server and, if successful, is then automatically signed on to further participating services and/or applications to which the user is entitled—based on the user's credentials.

(U) There are several problems with this model. The user must fully trust the authentication server, which may be problematic if the authentication server is managed by a second party, such as Microsoft. There also is the potential problem of basic security in that the authentication server is a single point of failure or central point of attack. Finally, there may be a privacy problem in that personal information could be collected as part of the authentication information.

(U) Note also that if the centralized authentication server were to be temporarily unavailable, a user would be precluded from accessing any additional target system during this period.

2.1.3.6.1.2.2 (U) Federated Model

(U) In general, as target systems become more numerous and as networks of systems become more complex, a centralized architecture becomes too complicated to manage efficiently. In this case, a federated architecture becomes more desirable. With federated authorization, credentials are propagated in a less centrally-controlled method than the original Microsoft Passport model. In addition, as the number of target systems (and even operating systems) proliferates, it is desirable that the SSO methodology be standards-based. There are currently three standards-based SSO techniques: Kerberos (via Tickets), PKI (via Certificates), and Security Assertion Markup Language (SAML) (via Assertions).(U) Since the GIG will have a broad geographic sweep in addition to a large number of interrelated participating organizations/partners, it is logical for the GIG to adopt a federated model for Single Sign-On implementation. The three candidates are described as follows:

2.1.3.6.1.2.2.1 (U) KERBEROS (Tickets)

(U) Kerberos is a password-based authentication protocol/mechanism that is based upon symmetric cryptography. A user’s password does not pass unprotected through a network subject to potential sniffing attacks by adversaries. Single sign-on can be implemented using Kerberos in the following manner as shown in Figure 2.1-8.

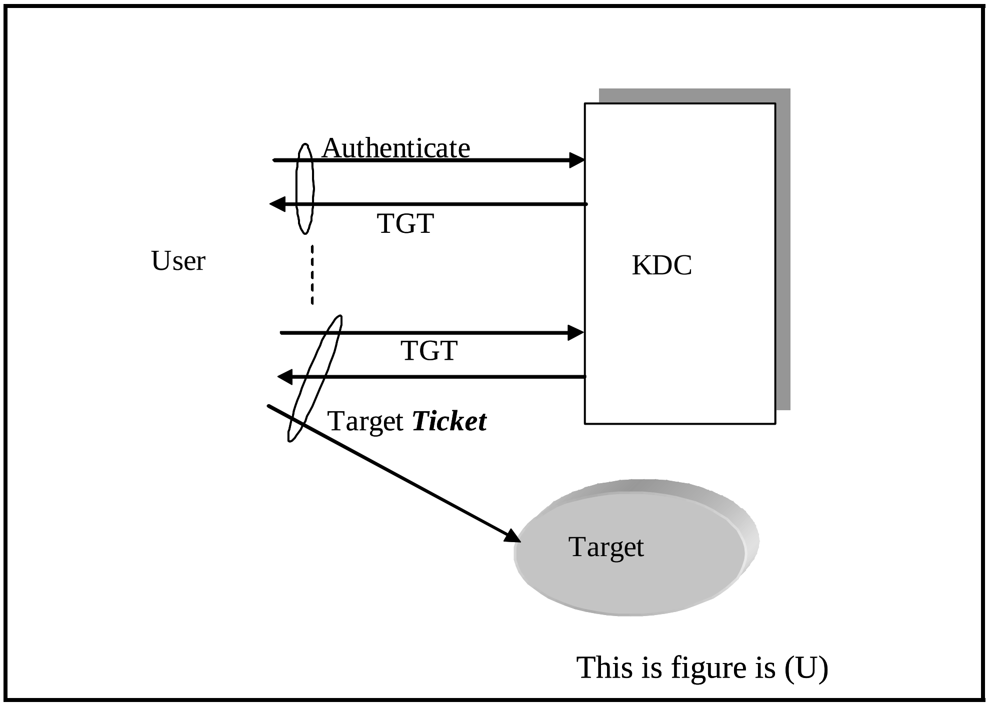


Figure 2.1-8: (U) Federated KEBEROS Based Single Sign-On

(U) Initially, a user would authenticate to a Key Distribution Center (KDC), which would in turn issue the user an encrypted Ticket Granting Ticket (TGT). For the lifetime of the TGT (typically several hours), the user is authorized to access a given target system by presenting the TGT back to the KDC. The KDC in turn then issues an enabling ticket that the user can present to the desired target system (without need for further authentication). Kerberos can be used across *Kerberized* platforms and/or applications. It is the standard inter-domain authentication protocol in Microsoft Windows .NET Server OSs and Windows 2000. Microsoft is updating its original basic Passport system using this model (Federated Microsoft Passport). One improvement is that a user can acquire a collection of target tickets and subsequently access a variety of target systems (within the ticket lifetimes), even if the KDC was to become unavailable due to an intervening system failure or KDC communication problems.

2.1.3.6.1.2.2.2 (U) PKI Certificates

(U) A SSO system based upon credential attributes, following the syntax defined by PKI X.509 certificates, is shown in Figure 2.1-9.

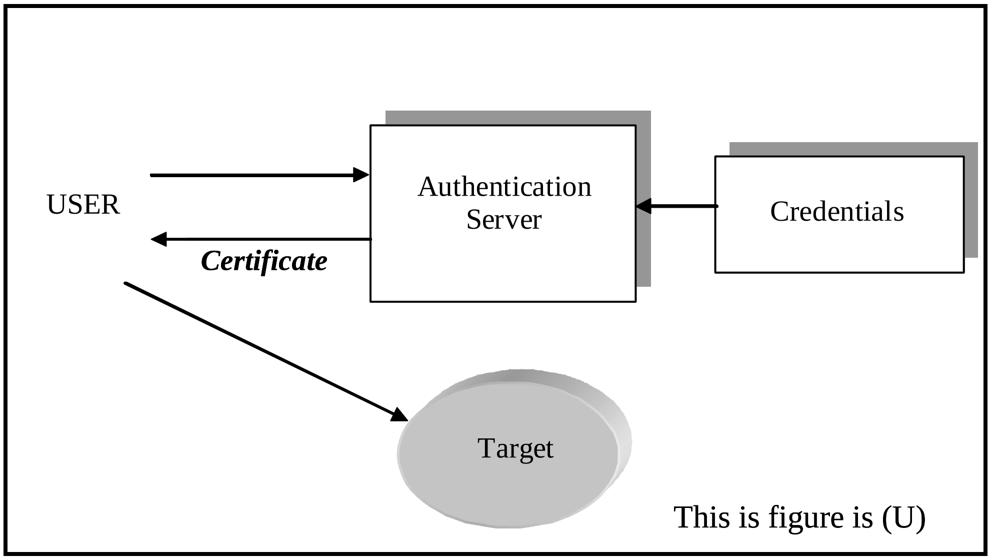


Figure 2.1-9: (U) Federated PKI-based Single Sign-on

(U) This model is federated in the sense that all the potential target systems are treated as equals in that they would each have assigned credential attributes defined within the SSO-enabling certificate, and the user may request access at any time to a pre-defined, included target system. When the user attempts to login to a candidate target system, it would forward its authorizing credentials held within an encrypted version of its attribute certificate. This certificate would have been signed by the original authorizing trust authority (using the private key of that authority), and it could be thus verified by the target system as authentic through use of the originating trust authority's public key. This application of digital signature technology thus enables the user to subsequently and transparently login to as many candidate target systems as are defined and allowed by the user's credential certificate.

(U) In addition to the certificate being digitally signed by the originating trust authority, it would be forwarded to candidate target systems in an encrypted format by using the public key of the target system. Any target system could then easily decrypt the *password* attributes through application of its own private key. As far as the user is concerned, all of the processing and transference of the attribute certificates would be done transparently in the background with the user simply accessing the target system and requesting use of available resources.

(U) Use of PKI-based asymmetric key technology could mesh nicely with the maturing DoD PKI and its supporting CAC smart card technology, which would retain the private key of each respective user.

2.1.3.6.1.2.2.3 (U) SAML (Assertions)

(U) Finally, an alternative SSO implementation may be based upon SAML as shown in Figure 2.1-10.

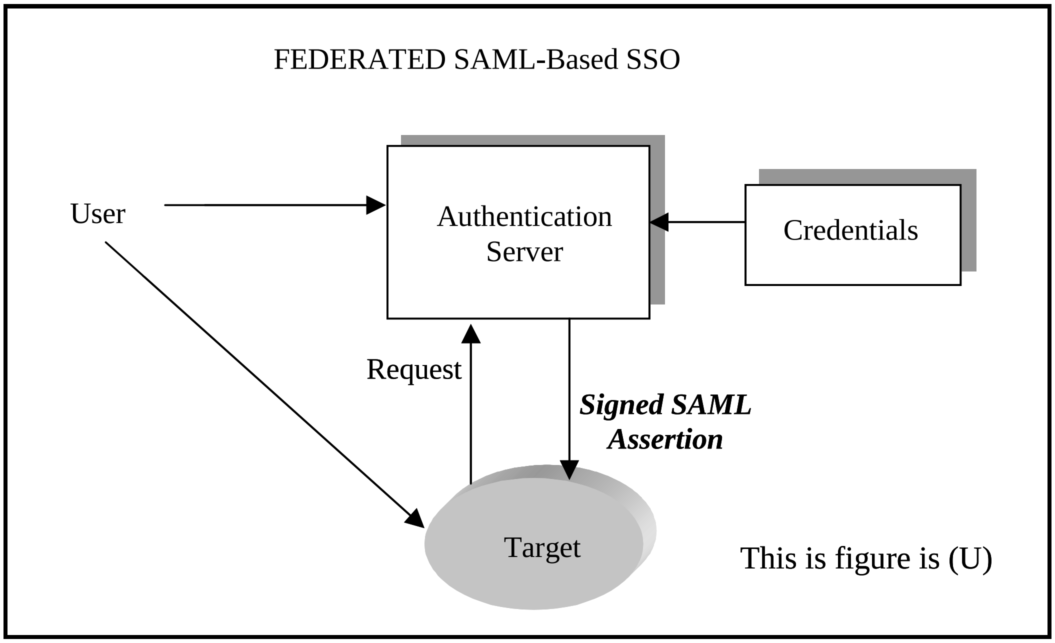


Figure 2.1-10: (U) Federated SAML-Based Single Sign-On

(U) Within a SAML-based SSO, the authentication server and all relevant target systems form a Circle of Trust to which a user may exercise SSO privileges. It is federated in the sense that the circle of trust is a predefined collection of target systems to which the user may potentially wish to apply the SSO mechanism. Each of the federated target systems is aware of the existence of the authentication server and knows how to request the signed SAML assertion when needed.

(U//FOUO) There are several examples of SAML being applied in projects in the DoD. One of these is the U.S. Navy's Space and Naval Warfare Systems Command (SPAWAR) Navy Enterprise Portal program, in which SSO capabilities based upon SAML are being introduced in order to tie together an estimated 200,000 applications on the Navy-Marine Corps Intranet (reached by 720,000 users distributed among active service members, civilian Navy employees, and contractors). In an initial demonstration, SAML-enabled SSO was provided to 5,500 users aboard the aircraft carrier USS Teddy Roosevelt.

(U//FOUO) Another example of SAML being used in DoD programs is the DISA/DIA (Defense Intelligence Agency) Virtual Knowledge Base (VKB) program. As is normally done with SAML implementations of SSO, this program uses the XML signature of the SAML assertions to provide for the non-repudiation of authentication/authorization credentials. In a prototype demonstration, the computation and processing burden of applying digital XML signatures was quite manageable and shown to be able to scale well to large user populations. This program also looked into the option of employing XML encryption of the SAML assertions in order to provide for confidentiality during transport. Unlike the XML signature experience, the XML encryption took much more computation time and was shown to not be amenable to scaling well to large populations. An alternative to using XML encryption would be to use the SAML implementation within established SSL/TLS (Secure Sockets Layer / Transport Layer Security) encrypted connections, since SSL is a proven and efficient protocol.

2.1.3.6.2 (U) Usage Considerations

2.1.3.6.2.1 (U) Advantages

(U) There are many clear advantages to SSO. For the individual user the benefits are highlighted by the convenience of not having to authenticate into each service that is accessed over the web (and having to remember a large number of passwords).

(U) In turn, SSO serves as a driver to the required supporting technologies of robust, multifactor-secure authentication (with biometrics, smart cards, etc.) by serving as the gatekeeper at the front end. It also provides a robustly implemented privilege management infrastructure, which keeps straight those net resources that a user can access through SSO.

2.1.3.6.2.2 (U) Risks/Threats/Attacks

(U) There were some disadvantages associated with the early versions of SSO technologies. For example, concerning password synchronization: while having a password synchronized across many applications may be more convenient for the user, it also results in a point of vulnerability. If a single password can be compromised, this compromises all applications linked to that password. This *risk aggregation* problem (clearly unacceptable in the GIG) is one of the key reasons why an earlier generation of so-called enterprise SSO products was not broadly adopted. Other factors that limited early adoption were the complexity and cost of deployment.

(U//FOUO) As the various SSO standards have been developed and deployed, a number of additional weaknesses were uncovered. These have led to revising and strengthening the underlying standard protocols. In 2000, D. Kormann and A. Rubin of AT&T Labs described weaknesses of the Microsoft Passport SSO protocol in their paper, “Risks of the Passport Single Sign-On Protocol” (See http://avirubin.com/passport.html). They identified three attacks on Passport: (1) Bogus Merchant Attack (where a user accesses a web site controlled by a malicious attacker who then proceeds to steal the user’s valuable authentication information), (2) Active Rewrite Attack, and (3) DNS (Domain Name System) Attacks. Requiring SSL security for all Passport exchanges would protect against the active rewrite attack. Similarly, adoption of DNSSEC enhancements (See http://www.dnssec.net/) would help to protect against DNS attacks.

(U) In 2003 SAML attacks were uncovered by T. Gross of IBM in “Security Analysis of the SAML Single Sign-On Browser/Artifact Profile” (See http://www.acsac.org/2003/papers/73.pdf). The attacks that were uncovered included Connection Hijacking / Replay Attack, Man-in-the-Middle Attack (by DNS spoofing), and HTTP Referrer Attack. Recommended solutions include use of secure channels such as SSL 3.0 or TLS 1.0 with unilateral authentication for all SAML-related message transfers. Clearly, as the various competing SSO protocols (Kerberos-based, PKI-based, or SAML-based) are implemented and studied, additional weaknesses and vulnerabilities may be discovered. This should only lead to strengthening the protocols as they are revised.

2.1.3.6.3 (U) Maturity

(U) Due to the increasing demands for enterprise-wide SSO capabilities, SSO technology has been maturing at a rapid pace over the past decade—pushed by the competitive pressures of the commercial marketplace. This has led to a variety of incompatible proprietary implementations, which has in turn led towards the desirable evolution of standards-based SSO architectures and protocols. Unfortunately, several distinct and incompatible islands of SSO standards have emerged (e.g., Kerberos, PKI, SAML), but there also has been a movement towards the interoperable merger of these standards so that truly universal and cross-platform SSO capabilities can emerge. In general, these individual technologies can be described as Mature (TRL 7 – 9).

2.1.3.6.4 (U) Standards

(U) The development of the various SSO architectures has been conducted in a number of formalized standards organizations and industrial vendor alliances. These are discussed below.

(U) There has been some movement towards the interoperability-enabling convergence of the various SSO standards protocols and their associated camps of supporting vendors. This is potentially advantageous to the evolution of the GIG, which should not be hindered by the adoption of security mechanisms that may eventually lose in the standards arena. One example of this convergence is work on defining SAML assertions in X.509-syntax attribute certificates. (See the privilege and role management infrastructure standards site at http://www.permis.org, and the NSF Middleware Initiative site at http://www.nsf-middleware.org/NMIR5/.) Another example of similarities between the PKI and Kerberos standards is that X.509 sign-on privilege attributes can be pre-defined with a validity period of hours or days, just like the Federated Kerberos-Based SSO architecture with its fixed lifetime tickets. This eliminates the need for the formalized revocation of X.509 attributes (as compared against the usually infrequent occurrence of revoking crypto keys in PKI X.509 public key certificates).

(U) It is also interesting to note that the Kerberos V5 version implements extensions to the original Kerberos protocol to permit initial SSO server authentication using public keys on smart cards. The original Kerberos protocol relied on symmetric secret key algorithms.

(U) Due to the continued success of each of the standards in its respective application domains, a mutual convergence of interoperability is preferable to conflict. For example, Kerberos is well known for certain applications and is supported by modern operating systems, whereas PKI certificate systems are widely spread (e.g., DoD PKI) and can provide portability across platforms.

(U) Large and influential vendors such as Microsoft, which has a history of supporting the WS-Federation, Kerberos-based SSO methodology, have introduced the concept of protocol transition. This is supposed to be a feature of Microsoft's Windows .NET Server and should allow a user to gain access to .NET Server-based resources by any one of a number of authentication mechanisms: Kerberos, PKI X.509 digital attribute certificate, SAML, etc. The target Windows .NET Server would then transition the sign-on token into a Kerberos ticket for use in the backend. This is an example of how, if provided with enough appropriate Inter Working Functions, a conglomeration of SSO standards can be made to interoperate successfully and securely.

2.1.3.6.4.1 (U) WS-Federation (Microsoft, IBM)

(U) The Kerberos-based SSO architecture has been championed primarily by Microsoft and its WS-Federation standard (promulgated jointly with IBM. See http://www-106.ibm.com/developerworks/webservices/library/ws-fed/). It is based upon the original IETF RFC 1510, "The Kerberos Network Authentication Service" by J. Kohl and C. Neuman (September, 1993), found at http://www.ietf.org/rfc/rfc1510.txt.

(U) Kerberos, developed at the Massachusetts Institute of Technology, is a system that depends on passwords and Data Encryption Standard (DES) symmetric cryptography in order to implement ticket-based, peer entity authentication service, and SSO access control service distributed in a client/server network environment. Kerberos came out of Project Athena and is named for the mythical three-headed dog guarding Hades.

(U) The overall Web Services Security Specification roadmap entitled "Security in a Web Services World: A Proposed Architecture and Roadmap" was promulgated by Microsoft and IBM in April, 2002. The base layer is called WS-Security, on top of which lie the layers of WS-Policy, WS-Trust, WS-Privacy, WS-SecureConversation, WS-Authorization, and WS-Federation (enabling SSO single sign-on). After development of these specifications, they were turned over to the non-profit OASIS standards body (See below).

2.1.3.6.4.2 (U) ITU

(U) The United Nations ITU-T standards organization (http://www.itu.int/home/) based in Geneva, Switzerland has been evolving its PKI-enabling X.509 standard into a standard that will support SSO-enabling attribute certificates.

2.1.3.6.4.3 (U) SAML (OASIS)

(U) The SAML v1.1 standard was approved and promulgated in September, 2003 by the Organization for the Advancement of Structured Information Standards (OASIS, at http://www.oasis-open.org). Webopedia defines SAML as "an XML (Extensible Markup Language)-based framework for ensuring that transmitted communications are secure. SAML defines mechanisms to exchange authentication, authorization and non-repudiation information, allowing SSO capabilities for web services." This allows organizations to create contractual federations and enables browsing end-users to reach services using a SSO with appropriate authentication/authorization information. SAML technology does not define any new authentication techniques itself, but rather merely enables the existing technology in XML. SAML is also targeted as a security services implementation to support Internet2.

(U) In order to foster the use of SAML as open source software, OpenSAML (http://www.opensaml.org/) has been developed. It is a set of open source Java and C++ libraries that are fully consistent with the formal SAML standard specifications. The OpenSAML toolkit may be licensed royalty-free from RSA.

2.1.3.6.4.4 (U) Liberty Alliance

(U) The Liberty Alliance "Project Liberty" (http://www.projectliberty.org/) was organized and introduced in 2001. It is a joint effort by 38 different companies, with Sun Microsystems as the motivating force. Also involved are staunch supporters of open source software such as the Apache Software Foundation and O'Reilly & Associates. Other involved technology companies include Verisign, RealNetworks, and Cisco.

(U) Liberty Alliance is adopting the SAML SSO architecture and protocols. Due to Sun Microsystems support of SAML, it is being applied in the Java sphere. The related Java technology API (Application Programming Interface) standard for SAML is covered by Java Specification Request JSR-155. (See http://www.jcp.org/.)

2.1.3.6.5 (U) Cost/Limitations

(U) While there are initial costs to implementing a robust and wide-reaching SSO capability, the eventual return on investment can be huge, and the realization of this is one of the prime drivers in persuading organizations to adopt SSO technology. When an automated and secure standards-based SSO system replaces a myriad of existing and disjoint independent traditional sign-on mechanisms, a tremendous administrative burden is lifted from the shoulders of both the individual user and the system administrator (e.g., help desks). A broadly adopted standards-based approach also allows for clearly defined evolution paths for SSO implementation.

2.1.3.6.6 (U) Dependencies

(U) Certainly one of the most important dependencies of a robustly secure SSO system is that a SSO architecture relies greatly on a very strong and secure multifactor initial user authentication, since if a malicious attacker were to successfully accomplish an invalid initial SSO login, they would effectively be given the keys to the kingdom of the violated authentic user (or one-stop shopping for hackers).

(U) The GIG thus is sure to benefit from a robustly developed and standards-based methodology of SSO. Fortunately, the evolution of SSO technologies is being driven by a number of strong commercial market forces. Specifically, there are three legislative processes that are requiring effective SSO capabilities in future commercial IT systems, particularly those dealing with sensitive—either personal or corporate proprietary—information.

- (U) Within the domain of corporate governance, the Sarbanes-Oxley Rule 404 requires public companies to centralize the reporting of who has access to what and who uses what. Moreover, business governance and privacy laws in many countries impose similar requirements.

- (U) Similarly, in the financial services market, the Gramm-Leach-Bliley Act specifies the need for stronger audit and separation of duties, in order to control who, how, and when users access information and systems.

- (U) Finally, the healthcare market is a primary revenue-driving segment for many SSO vendors. The Health Insurance Portability and Accountability Act (HIPAA) requirement for an audit trail that associates information access to individual identities becomes mandatory in April, 2005. Healthcare typically involves the deployment of workstations that need to be accessed by many healthcare workers, who must frequently and quickly log in and out of these systems. A robust and secure SSO technique will be very beneficial to this requirement.

2.1.3.6.7 (U) Alternatives

(U) The alternative to implementation of an integrated SSO infrastructure within the GIG is to continue the operation of disparate and independently maintained and administered SSO mechanisms for each application or resource that GIG users will want to use. A partial solution, which could be application sensitivity-based in that SSO capability, could be developed for most of the GIG-spanning resources. However, certain very sensitive (e.g., command and control-oriented) applications may require independent and rigorously assured authorization and authentication every time they are accessed. As the GIG-wide SSO solution and supporting privilege delegation infrastructure matures, the scope of its applicability may indeed expand.

2.1.3.6.8 (U) References

(U) Simple Authentication and Security Layer, http://asg.web.cmu.edu/sasl/.

(U) "Single Sign-On and the System Administrator", by M. Grubb and R. Carter (Duke University), LISA'98 conference, http://www.usenix.org/publications/library/proceedings/lisa98/full_papers/grubb/grubb.pdf , 6-11 December 1998.

(U) "Single Sign-On Architectures", presentation by Jan De Clercq (HP), http://www.esat.kuleuven.ac.be/cosic/seminars/slides/SSO.pdf, 2002.

(U) "Identity Management: A Technical Perspective", presentation by Richard Cissee,

http://www.calt.insead.edu/fidis/workshop/workshop-wp2-december2003/presentation%5CTUB%5CTUB_fidis_wp2_workshop_dec2003.ppt, December 2003.

(U) "Single Sign-On Across Web Services", presentation by Ernest Artiaga, CERN OpenLab Security Workshop, http://openlab-mu-internal.web.cern.ch/openlab-mu-internal/Documents/Presentations/Slides/2004/04-09_EA_Security_Wokshop-SingleSignOn.ppt , April 2004.

(U) "Navy Deploying Its Battle Plan: SAML", by Anne Chen, http://www.eweek.com/article2/0,1759,1502403,00.asp, 20 October 2003.

(U) "Lessons Learned in a Department of Defense Program (The Virtual Knowledge Base VKB)", presentation by Kevin Smith, http://www.omg.org/news/meetings/workshops/Web%20Services%20USA%20Manual/02-3_K_Smith.pdf.

(U) "Web Services Security", presentation by Sang Shin (Sun Microsystems), http://www.javapassion.com/webservices/WebServicesSecurity4.pdf, January 2004.

(U) "SAML Basics", presentation by Eve Maler, http://www2002.org/presentations/maler.pdf, 2002.

(U) "Survey of the Status of Security and Emerging Security Innovations for Key Technological Protocols, Recommendations, Specifications and Standards Used in E-commerce", by Angela Mozart, http://www.giac.org/practical/GSEC/Angela_Mozart_GSEC.pdf, November 2003.

(U) "Gartner report: Password Management, Single Sign-On, and Authentication Management Infrastructure Products: Perspective", by Ant Allan, 7 January 2003.

2.1.4 (U) I&A Gap Analysis

(U//FOUO) Gap analysis for the Identification and Authentication Enabler indicates that the main areas of required future development are as follows:

- (U//FOUO) Complete the development of Protection Profiles for Medium and High Assurance authentication technologies (e.g., biometrics).
- (U//FOUO) Develop an authentication framework standard that includes SoM levels, authentication session scoring, and a SoM forwarding structure.
- (U//FOUO) Develop a standard for the methods/protocol of remote access point retrieval of authentication privileges.
- (U//FOUO) Develop a token with onboard biometric and liveness test (to assure that automated logon is not taking place), or offboard biometrics (communicated to token). Candidate offboard biometrics are iris scan, retinal scan, face recognition, hand geometry, voice recognition, etc. Based on current technology, only a thumbprint/fingerprint reader could be integrated directly onto a smart card token.
- (U//FOUO) Develop a high assurance DoD PKI Class 5 token w/Type I cryptography (where definition of Class 5 token is for use with classified information + hardware token + using Type I cryptography + having assurance/trust in security critical functionality throughout its lifecycle, including design, development, production, fielding, and maintenance).
- (U//FOUO) Develop a scalable re-authentication scheduling algorithm, adjustable per sensitivity of application, access location, and user profile.
- (U//FOUO) Develop a scalable authentication server that is able to interpret and use I&A session scores and comply with the GIG authentication standards. The server function will need to be secure, efficient, accurate, and transparent in terms of performance impact. In addition, it should operate in multiple architectural constructs (e.g., in-line, embedded, co-processor, remote).
- (U//FOUO) Develop an Identification Registration/Management Infrastructure that can support all GIG customers (DoD, IC, and all temporary/permanent partners).
- (U//FOUO) Develop a common GIG-wide Single Sign On mechanism, protocol, and architecture.
- (U//FOUO) Develop a GIG standard for authentication confidence metrics.

(U//FOUO) In addition, the following gaps must be satisfied under other IA System Enablers that directly support this IA System Enabler

- (U//FOUO) Develop converged standards for Partner Identity Proofing, enabling identity interoperability with future GIG partners (e.g., allies, coalition partners, civil government, DHS). (See Section 2.7, Management of IA Mechanisms and Assets)

- (U//FOUO) Develop a common identification management and ID proofing standard for all future GIG entities (human users, devices, processes). (See Section 2.7, Management of IA Mechanisms and Assets)
- (U//FOUO) Ensure metadata standard includes the capability for binding authenticated sources to GIG information. (See Section 2.2, Policy-Based Access Control)

(U//FOUO) Technology adequacy is a means of evaluating the technologies as they currently stand. This data can be used as a gap assessment between a technology's current maturity and the maturity needed for successful inclusion in the GIG in 2008.

(U//FOUO) The following two tables list the adequacy of the Identification and Authentication technologies with respect to the enabler attributes discussed in the RCD. Not shown in the tables below are entries for Authentication Protocols which are in general quite adequate, in so far as their strength and flexibility is concerned.

Table 2.1-3: (U) Technology Adequacy for Tokens and Biometrics

This Table is (U)

		Technology Category		
		Tokens	**Biometrics**	**Required Capability (attribute from RCD)**
Enabler Attribute	**Standard**			IAAU3, IAIR2, IAIR4
	Secure Solution			IAAU1, IAAU3, IAAU8, IAAU9, IAAU18, IAAU19, IAAU20, IAIR1, IAIR6
	Scalable Solution		N/A	IAAU10, IAAU23, IAIR2, IAIR5, IAIR6
	Protection Profile			IAAU1
	High Assurance			IAAU2, IAAU24
	Distributed/ Global Reach		N/A	IAAU1, IAAU6, IAAU17, IAAU21, IAIR2
	Verifiable Solution			IAAU1, IAAU12-IAAU15, IAIR1, IAIR3, IAIR4, IAIR5

This Table is (U)

Table 2.1-4: (U) Technology Adequacy for Single Sign-On and Authentication

This Table is (U)

		Technology Category			
		Single Sign On	**Authentication Confidence**	**Device Authentication**	**Required Capability (attribute from RCD)**
Enabler Attribute	**Standard**				IAAU4, IAAU5, IAIR1, IAIR7
	Secure Solution		N/A		IAAU8, IAAU22, IAIR6
	Scalable Solution		N/A		IAAU23, IAIR6, IAIR7
	Protection Profile	N/A	N/A		
	High Assurance		N/A		IAAU22
	Distributed/ Global Reach				IAAU6, IAAU25, IAAU23, IAAU21, IAAU17, IAIR7
	Verifiable Solution		N/A		IAIR1

This Table is (U)

2.1.5 (U) Identification and Authentication: Recommendations and Timelines

(U) The following is a list of preliminary recommendations for advancing the technologies required for the successful implementation of this GIG enabler:

- (U//FOUO) Define a converged Partner Identity Proofing standard that has been vetted and accepted by partner communities.
- (U//FOUO) Develop a common GIG-wide device/service authentication techniques and standards, due to the relative immaturity of this technology area.
- (U//FOUO) Rapidly advance research into the relatively new area of authentication confidence metrics.
- (U//FOUO) Develop a scalable, robust, and distributed authentication server capability.
- (U//FOUO) Develop an accepted high assurance biometric authentication technique.
- (U//FOUO) Assure ongoing and future developments of the DoD CAC Common Access Card will support all future GIG requirements (including Class 5 token).
- (U//FOUO) Advance the selection of a GIG-wide architecture for Single Sign-On (from the candidates described in this document, such as SAML-based or PKI-based). Include in this process the complete analysis of the proposed NCES single sign-on architecture.

(U//FOUO) Figure 2.1-11 contains preliminary technology timelines for this IA System Enabler. These are the result of research completed to date on these technologies. As the Reference Capability Document and the research of technologies related to these capabilities continue, these timelines are expected to evolve.

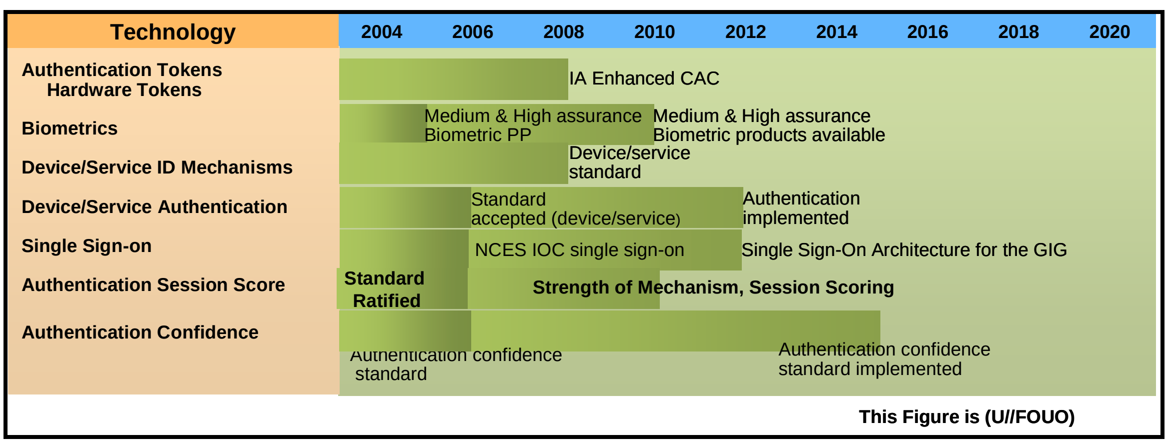


Figure 2.1-11: (U) Technology Timeline for Identification & Authentication

2.2 (U) POLICY-BASED ACCESS CONTROL

(U//FOUO) Policy-Based Access Control is the use of flexible, hierarchical rules to determine whether to grant or deny access to GIG assets at points throughout the GIG. This policy-based access control capability is also distributed. It provides common GIG access control services across the enterprise, supports an enterprise wide digital access policy, and provides decision processing location transparency to the user to improve availability and load sharing capability. GIG assets include all resources within the enterprise, such as hardware (e.g., routers, servers, workstations, security components), software (e.g., services, applications, processes), firmware, bandwidth, information, and connectivity.

(U//FOUO) From a context prospective, today's information sharing capabilities are not sufficient to support the net-centric operations vision. Current information sharing is far too constrained through:

- (U//FOUO) A culture that fosters not sharing
- (U//FOUO) Physically separate, system-high environments
- (U//FOUO) Limitations of information assurance (IA) technology to safely support assured information sharing

(U//FOUO) Our no-risk culture allows access to classified information only to recipients who have the proper clearance and a need-to-know. But this accessibility culture must change to support the vision of information sharing functionality that empowers users through easy access to information, anytime, anyplace, and anywhere in support of operational requirements with attendant security.

(U//FOUO) The GIG information sharing philosophy is fundamentally different as it is a sharing centric security philosophy. The user is presented with information consistent with such factors as his security clearance, operational situation, privilege and policy, then decides what information is needed and pulls that information. This differs from the need-to-show paradigm in which the data originator decides to whom to provide the data (i.e., no one else knows the data exists).

(U//FOUO) Policy-Based Access Control supports this need to share paradigm and represents a transformation of historical mandatory and discretionary access control. It considers security risk and operational need as part of each access control decision. It thus recognizes that situational conditions (e.g., peacetime, war, terror threat levels, location of people) will drive the relative weight of operational need and security risk in determining access.

(U//FOUO) The access control decisions can adapt to varying situational conditions in accordance with an access control policy. Each policy prescribes the criteria for determining operational need, the acceptable security risk, and the weighting between the two under various conditions. Thus the model can support extremely restrictive policies and also those that provide the widest sharing under specific conditions with added risk. This new access control model has been named Risk Adaptable Access Control (RAdAC).

2.2.1 (U) GIG Benefits due to Policy-Based Access Control

(U//FOUO) The Information Assurance constructs used to support Policy-Based Access Control provide the following services to the GIG.

- (U//FOUO) Provides standardized access control behavior for information, communications, and services throughout the GIG
- (U) Provides fine-grained access control based on the labeled value and life cycle constraints of the information
- (U) Provides fine-grained access control based on the privileges and priority of the user (user is defined as a human user, entity, or service)
- (U//FOUO) Provides ability to segregate multiple communities sharing the GIG to increase availability while providing dynamic connectivity as needed
- (U//FOUO) Supports Single Sign-on (SSO) because an authorization granted is then recognized throughout the GIG
- (U) Allows flexibility to tailor aspects of enterprise policies by region, COIs, C2 Node, etc.
- (U) Supports data owner information life cycle policy to track and control object creation, dissemination, use, and destruction

2.2.2 (U) Policy-Based Access Control: Description

2.2.2.1 (U) Core RAdAC Functions

(U//FOUO) Policy-Based Access Control is a critical enabler for sharing information and services within the GIG. Access Control checks will no longer follow the traditional check for an exact match of mandatory (e.g., credentials) and discretionary (e.g., privileges) checks. Instead, the RAdAC Model will be employed. RAdAC is a rule-based access control policy, based on real-time assessment of the operational need for access and the security risk associated with granting access. Figure 2.2-1 depicts the RAdAC model. There are two core functions within RAdAC, Security Risk Determination and Operational Need Determination.

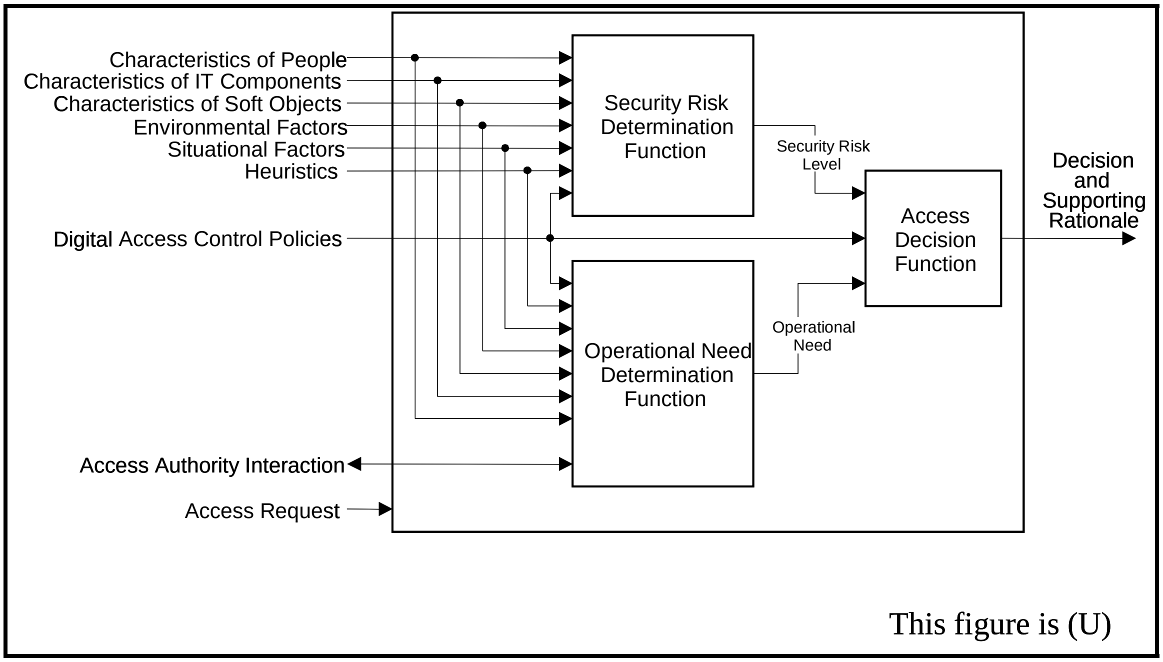


Figure 2.2-1: (U) RAdAC Functional Model

(U//FOUO) Security Risk Determination provides a real-time, situational, and probabilistic determination of the security risk associated with granting the requested access. The challenge here is to come up with ways to quantitatively express risk. The security risk for granting the access will be determined for at least three different areas:

- (U//FOUO) The person receiving the information
- (U//FOUO) The IT components the person is using
- (U//FOUO) Those that will otherwise be involved in sharing the information

(U//FOUO) Operational Need Determination assesses the operational need of a requestor to access some information. A person's membership in some COI or organization, their rank or role in an organization, their location, or a supervisor's approval might all be contributing factors to establishing their need to know information, but ultimately access control policy will specify how to use these factors to determine operational need.

(U//FOUO) An important attribute of Operational Need Determination is the capability of allowing an exception to an access control decision. The access control policy would specify who is entitled to approve an exception. For example, a commander may determine particular data is critical to his mission and grant access to data to which his forces would normally not have access. However, the policy must grant the commander this right.

2.2.2.2 (U) Assured Metadata and Data Describing Enterprise Elements

(U//FOUO) Assured metadata and data describing enterprise elements such as users, IT components, environment, and situation serve as inputs to the RAdAC functional model. Not all inputs may be required to make a specific access decision. Digital access control policy will dictate the minimum decision criteria and how limited input affects the access control decision.

- (U//FOUO) Characteristics of people who create and consume information will be used to measure their risk and to determine their operational need. These characteristics might include identifier, citizenship, security clearance level, and source of clearance, organization, COI membership, military rank, length of service, current operational assignment, job title, GIG system privileges—and any other characteristics that might be usable in determining their security risk and operational need. Characteristics of the authentication process that granted a person access to the system would also be included here since multiple proofs of identity increase how certain the system is concerning the true identity of a requester.

- (U//FOUO) Characteristics of IT components that create information and enable users to create, share, and use information will be used to determine security risk. Determining the robustness of the components is the primary consideration. Therefore, such things as identifier, operating system, hardware platform features, current configuration conformance to certified configuration, third-party robustness evaluation, owning organization, system administrator characteristics, connectivity to unprotected networks, and software distribution protection might be characteristics considered when determining the risk associated with IT components. Furthermore, the operation of these components as a system must be considered.

- (U//FOUO) Characteristics of Soft Objects contribute to the access decision, affecting both the security risk measurement and the determination of operational need. Soft objects include data, applications, and services.

- (U//FOUO) The important characteristics of an object being accessed might include its identifier, source/originator or controlling entity (including COIs), a description of the type of data and its value, a description of the data source and its pedigree, intended roles and expected uses of this object, object life cycle properties, and traditional labeling information. Object life cycle properties include object-level attributes that constrain use, dissemination, and disposition after use.

- (U//FOUO) Traditional labeling information would include such data as classification level, releasability, and caveat handling. The metadata will be cryptographically bound to the data to which it applies, so the requestor can validate the authenticity of the data.

- (U//FOUO) Environmental factors apply to people, IT components, and objects, and can be used in determining both security risk and operational need. Environmental factors include such things as a physical location and any adversarial threat associated with that location. The adversarial threat should be tied to the GIG operational threat model and risk assessment. It might indicate for a particular location—or class of locations—the probability that a specific threat or attack could happen. Location might also be a factor in determining operational need. All GIG users in a particular location, such as Iraq, might have a need to access some specific class of information.

- (U//FOUO) Situational factors are national, enterprise-wide, or local indicators of some situational condition that might affect access control decisions. The terrorist threat level, for example, might be used to change criteria for determining operational need. For example, an indication that the enterprise is under cyber attack or nuclear attack might be other such situational indicators that could affect access.

- (U//FOUO) Heuristics are intended to represent the knowledge of the information sharing system that it has acquired from past information sharing and access control decisions. User-based heuristics might capture previously granted object access and can be used to help assess current risk and weigh operational need for future similar access requests. System-based heuristics may capture knowledge of compromises that have resulted under various access conditions in order to refine policy to avoid similar future compromises. A policy must specify the degree to which heuristics should be considered in each access decision.

2.2.2.3 (U) Digital Access Control Policy

(U//FOUO) Digital access control policies will be the key to making the RAdAC model successful. They must be capable of specifying the policy for each step of the access control process. They must also be capable of expressing rules for various types of access such as discovery, retrieval, modification, and execution rights. In other words, the requestor may be able to discover the object/service, but may not have rights to access the data without verification of need to know.

(U//FOUO) A policy would also be conditional in nature. It could stipulate different rules of access depending on the current operational condition or mission need. An example condition might be the current DEFCON level. Under one condition, access might be limited to those within a COI, while under another condition those with special operational needs might be given access. Policy flexibility is crucial.

(U//FOUO) Another aspect of digital access control policies is that multiple policies will exist in the GIG. There will be enterprise level policies and local policies (e.g., COI policies). The composite set of policies that apply to the object/service will be enforced during access control checks.

(U//FOUO) For access control to meet the information sharing needs of the GIG, digital access control policy must extend beyond the initial RAdAC decision through the inclusion of object life cycle attributes that accompany the soft object. For example, these attributes will specify whether the entity can save or print or forward the object, whether it is provided as read only, when the object's lifetime will expire, and what methods are acceptable for secure disposal of an object.

2.2.2.4 (U) IA Enabler Dependencies

(U//FOUO) Identification & Authentication. The authenticity of requester can be measured through the robustness and number of authenticators used to validate the requester's identity. Periodic re-authentication may be necessary for a I&A Strength of Mechanism (SoM) score to be considered viable by the RAdAC model.

(U//FOUO) Protection of User Information. This environment will be a significant factor in calculating security risk since it is a major portion of the Characteristics of IT Components input.

(U//FOUO) Dynamic Policy Management. Digital access control policy will be a subset of the policies managed dynamically in the GIG. The distributed RAdAC function will require the distribution, synchronization, and revocation capabilities offered by the Dynamic Policy Management environment.

(U//FOUO) Network Defense and Situational Awareness. RAdAC policy depends upon the enterprise's Information Condition (INFOCON) and threat levels on suspected or actual Information Warfare attack as a subset of its Situational Factors input.

(U//FOUO) Management of IA Mechanisms and Assets. RAdAC will depend upon this enabler to assure use of specific routes that guarantee Quality of Protection, management enforcement of IT Components with their approved uses and configurations, and certification & accreditation of enterprise domains as a risk input.

2.2.3 (U) Policy-Based Access Control: Technologies

(U//FOUO) For simplicity, the discussion of technologies for Policy-Based Access Control is divided into three sections:

1. (U//FOUO) Core RAdAC that addresses the internal computation of risk and operational need
2. (U//FOUO) Assured Metadata that supports RAdAC decision-making and enforcement
3. (U//FOUO) Dynamic Policy that influences RAdAC decision-making and enforcement

2.2.3.1 (U) Core RAdAC

(U//FOUO) The core RAdAC functions of security risk and operational need determination are very new ideas in the access control sphere, both in industry and Government. Traditionally, both these functions have been handled as administrative procedures that are then implemented and enforced through a combination of physical access controls (e.g., locked or guarded facilities) and static, but modifiable, logical access control business rules (e.g., traditional discretionary access controls in mainstream operating systems and mandatory access controls in multilevel environments). These static business rules can be correctly referred to as access control policy, but the underlying technology essentially assesses a request against a list of authorized actions and provides a binary allow/disallow decision to an enforcement mechanism.

2.2.3.1.1.1 (U) Technical details

(U//FOUO) IT security risk has historically been a calculation (either qualitative or quantitative) of the loss expected due to an attack being carried out against a valuable asset with a specific vulnerability. The exposure of the asset through the vulnerability and the probability the attack will occur are significant inputs for the final calculation. While technologies exist to guide a security professional in performing this type of risk assessment for a business or system, applying this technique to the access control domain is a very new idea.

(U//FOUO) In the access control domain, soft objects are the information assets that can be exposed to threats in the environment within a specific situation (including users) through vulnerabilities in the IT Components themselves. This relationship indicates that most of the RAdAC inputs affect security risk determination in one way or another—as described below. A high-level analysis of these RAdAC inputs shows that most will be textual in nature.

- (U//FOUO) Availability and integrity risk - Characteristics of IT components influence whether these tenets of IA would be placed at risk if access is authorized. For example, authorizing the release of a 40GB imagery file through a 28kbps tactical circuit would effectively cause a sustained denial of service for all users of that tactical circuit.

- (U//FOUO) Aggregation - Situational Factors should include details of what information is already available at a user's IT platform to assess the risk of aggregation (multiple Unclassified documents being combined to learn Classified information). As multiple services are subscribed to by a single user, the risk of aggregation (multiple unclassified inputs = classified information) increases.

- (U//FOUO) User information and platform context - Consideration of the classification of current information on the user's IT platform should be considered alongside the capabilities and assurances of the user's platform. For example, if a cleared user is subscribed to all FOUO services and requests a classified document, the risk of disclosure increases greatly if the platform cannot support MLS or MILS processing.

- (U//FOUO) Identity factors - Clearance and formal access approvals of the user, assurance of the user's identity, and assurance of bindings to roles and COIs are critical factors to determining risk.

- (U//FOUO) Classification lifetime - Classified lifetime of a Soft Object is an important consideration for risk. If declassification is expected within hours versus years and the specific operation that the information pertains to is already underway, the risk of disclosure to an uncleared soldier is much lower than it ordinarily would be.

- (U//FOUO) Violation of traditional access models - All things considered equal, any access that violates the Bell-LaPadula properties[3] should sharply raise the risk value.

- (U//FOUO) Probability of overrun - Risk of disclosure should increase due to the proximity of enemy forces and probability of overrun. This should be captured in the Environment Factor.

- (U//FOUO) Unavailable input parameters - Lack of input parameters (e.g., no value for IT environment) or low reliability of input parameters (e.g., non-authoritative source provides input for an IT environment segment) should increase the resulting risk due to unknowns.

- (U//FOUO) Heuristics - Heuristics from previously authorized similar requests (proximity with respect to time or content) should result in a reduced security risk.

- (U//FOUO) Transitivity - There are transitive security risks to consider in a highly-connected environment when authorization exceptions are permitted. Authorizing a classified document to one member of a COI operating at an unclassified level has implications that reach beyond that individual User making the access request.

- (U//FOUO) External connections - Since policy negotiations between security domains is a desirable dynamic policy feature, there is a potentially higher risk that all information released to an external domain should carry. Domain interconnection only begins to scratch the surface of risks associated with interconnections within the GIG.

- (U//FOUO) Enterprise C&A - GIG risks associated with IT Components within the enterprise must be considered in RAdAC risk determination. With the direction DIACAP is heading, near real-time knowledge of GIG system's risks, countermeasures applied to them, and residual risk that is accepted by a cognizant approval authority will be available through the eMASS system. The RAdAC model should interface to the eMASS services to understand residual risk in systems involved in the access path. This data should be presented to RAdAC via

[3] (U) The **Bell-Lapadula Model** of protection systems deals with the control of information flow. It is a linear non-discretionary model.

the "Characteristics of IT Components" and "Environment Factors" inputs.

- (U//FOUO) Identity Strength of Mechanism - A higher authentication robustness (e.g., a 3-factor authentication versus 2-factor) should yield a lower risk score.

- (U//FOUO) Soft Object Life Cycle Characteristics - Soft Object characteristics that limit or preclude widespread dissemination should raise the risk score, and imposed life cycle characteristics on a specific instance of information such as "do not copy, do not print, do not further disseminate" may reduce the risk of disclosure.

(U//FOUO) The other major function of the core RAdAC model is operational need determination, a function somewhat understood in the administrative domain and much less understood technologically. Outside of workflow technology that retrieves a manager's approval for need-to-know, no technology exists to perform this function. Characteristics of IT Components will have little to no impact to this function, and Situational Factors and Heuristics will probably have the most impact.

2.2.3.1.1.2 (U) Usage considerations

(U//FOUO) The successful usage of core RAdAC as the GIG access control model will require substantial proof of correctness, a highly robust distributed design, low-latency performance, life cycle information management, and significant buy-in from the various GIG user communities. The shift to a need-to-share philosophy is essential but largely depends on the assurances that the technology can mitigate risks associated with doing so.

(U//FOUO) For RAdAC to be successfully deployed and used throughout the GIG, the existence of any alternate access control mechanisms is problematic. Part of the RAdAC environment description must address how RAdAC is always invoked and non-bypassable within the enterprise. This description contributes to the proof of correctness needed to gain customer acceptance of the technology.

2.2.3.1.1.2.1 (U) Implementation Issues

(U//FOUO) Since most of these inputs are textual, RAdAC risk determination should be performed using technology that can parse, understand meaning, and reason about relationships under an imposed policy. Otherwise, the performance impact of translation between text and numeric scores will prove very costly, and RAdAC risks being inflexible in accommodating more than one ontology.

(U//FOUO) The ontology problem for textual inputs is very significant. In a trivial case, consider the existing U.S. Air Force and U.S. Navy ontologies used daily. A user identified with the rank of Captain in the Air Force is an O-3, who is a junior officer compared to a Navy Captain, who is a senior O-6 typically assigned to commander roles. Operational Need determination should weigh an Air Force Captain's verification of an E-5's need to know as less than a Navy Captain's verification of an E-5's need to know. A technology that doesn't understand more than one ontology cannot understand these distinctions that can be critical in determining access control risk and operational need.

(U//FOUO) To comply with national laws that strictly prohibit disclosure of classified information to users without appropriate clearances, any immediate implementations of RAdAC must implement a mathematic model to prove correctness for handling classified information. To comply with the national law in the near term, this model should map to traditional Discretionary and Mandatory Access Control (DAC and MAC) models, and it must never violate the properties established by the Bell-LaPadula confidentiality model.

(U//FOUO) Commercial access control technology is not heading in the direction of risk calculation. Rather, industry understands the traditional access control models of DAC and MAC. Role-based Access Control (RBAC) has recently reached maturity, and Attribute-based Access Control (ABAC) is just beginning to mature. Because of this, the scope of RAdAC may be more suited to the service-oriented architecture domain rather than the operating system domain so that both sets of access control models can coexist.

(U//FOUO) RAdAC must be able to offer performance guarantees despite the complexity of its calculations and the varied inputs required to make a decision. RAdAC must also be deterministic and produce an access control decision for every request.

(U//FOUO) RAdAC must provide decision rationale to support appeals for operational need-to-know, audit, and heuristics-based learning.

(U//FOUO) Heuristics implementation can take the form of either user-based or system-based knowledge of past actions, and most likely both are needed. In either form, the heuristics data must be verifiably system-recorded (not spoofed or modified) and rapidly available to the RAdAC decision service. Heuristics is a desirable RAdAC feature that is not as crucial as other features and can be delayed until later increments.

(U//FOUO) RAdAC's distributed model must be able to support the dismounted soldier with intermittent connectivity in addition to the CONUS-based desk user and the enterprise service tier. This distribution model should be able to synchronize updates to access control policy and information needed to make decisions to support operations in an offline mode.

(U//FOUO) RAdAC requires assured metadata about Soft Objects and assured data for its other inputs to make an informed decision and protect itself against well-known security threats. This assured metadata must be tightly bound to the information it describes and must itself have verifiable integrity.

(U//FOUO) RAdAC must provide state management to detect and consider repeated failed access attempts. This state management needs to be extremely lightweight to scale well in order to support thousands of users.

2.2.3.1.1.2.2 (U) Advantages

(U//FOUO) The RAdAC concept offers the following significant advantages relative to traditional access control schemes.

- (U//FOUO) Supports GIG need-to-share vision through dynamic access control decision-making that weighs security risk and operational need versus traditional hard-coded access control

- (U//FOUO) Allows broader scope of inputs that contribute to access control decision-making, including operational need and situational urgency
- (U//FOUO) Provides fine-grained access decisions (not just “allow” or “disallow”) that specify required transport path or object life cycle attributes to secure the risk of granting access

2.2.3.1.1.2.3 (U) Risks/Threats/Attacks

(U) The primary risks to RAdAC are:

- (U//FOUO) Spoofed or altered RAdAC inputs which can allow unauthorized access
- (U//FOUO) Access DoS attacks (counter detailed in CND RCD section) which prevent authorized access by legitimate users
- (U//FOUO) RAdAC bypass (direct object access)
- (U//FOUO) Distributed environment synchronization attacks

2.2.3.1.1.3 (U) Maturity

(U//FOUO) Both security risk and operational need determination technologies are in the conceptual stage. Basic principles have been observed and reported in the Assured Information Sharing Model white paper, and practical applications are being explored through a separate study. Technology maturity is rated as Early (TRL 1-3).

2.2.3.1.1.4 (U) Standards

(U//FOUO) Potential standards that loosely apply include:

Table 2.2-1: (U) Access Control Standards

This Table is (U)	
Standard	**Description**
Role-Based Access Control (ANSI INCITS 359-2004)	Describes Role Based Access Control (RBAC) features that have achieved acceptance in the commercial marketplace. It includes a reference model and functional specifications for the RBAC features defined in the reference model
Validated Common Criteria protection profiles	For access control, including Controlled Access Protection Profile, Labeled Security Protection Profile, Role-Based Access Control Protection Profile
Multinational Information Sharing Environment Protection Profile v.1.0	Contains functional and security requirements for sharing information up to Secret among multinational partners

(U//FOUO) Potential supporting commercial technologies include:

Table 2.2-2: (U) Technologies Supporting Access Control

This Table is (U)

Standard	Description
Security Assertion Markup Language (SAML) v2.0	The Security Assertion Markup Language (SAML) is "an XML-based framework for exchanging security information. This security information is expressed in the form of assertions about subjects, where a subject is an entity (either human or computer) that has an identity in some security domain. (W3C standards organization)
eXtensible Access Control Markup Language (XACML) v1.0	OASIS Extensible Access Control Markup Language (XACML) defines a core schema and corresponding namespace for the expression of authorization policies in XML against objects that are themselves identified in XML. (OASIS standard 6 Feb 2003; a working draft of v2.0 is available)
DARPA Agent Mark-up Language (DAML)	Provides constructs to create ontologies and metadata markup information for machine readability
Web Ontology Language (OWL) v2.0	Provides a language that can be used to describe the classes and relations between them that are inherent in Web documents and applications.
Web DAV Access Control Protocol (RFC3744)	WebDAV stands for "Web-based Distributed Authoring and Versioning". It is a set of extensions to the HTTP protocol which allows users to collaboratively edit and manage files on remote web servers. (IETF standards organization)
Content Based Information Security	Joint Forces Command-sponsored advanced technology concept demonstration that supports the notion of abstracting the complexity of label development from the operator through the use of roles. It also supports the notion of a hierarchy of policies to control sharing.

This Table is (U)

2.2.3.1.1.5 (U) Costs/limitations

(U//FOUO) A large monetary cost will be incurred to design, develop, test, and field RAdAC into the GIG enterprise since there is no similar commercial technology.

(U//FOUO) A significant performance cost will be associated with access control decision-making due to the quantity of RAdAC model inputs and the amount of detail required for these inputs. Current access control technologies compare a request against a user's identity—and an associated list of authorizations—and then produce a binary access decision. The complexity of RAdAC will most likely increase the computation needs for each decision by an order of magnitude.

(U//FOUO) There will also be significant network bandwidth cost due to the transfer of RAdAC inputs and outputs and the distribution of RAdAC heuristics, although the distributed design can be optimized to reduce the bandwidth cost.

2.2.3.1.1.6 (U) Dependencies

(U//FOUO) Implementation of the RAdAC concept relies on several technologies covered by other IA System Enablers:

- (U//FOUO) Access Control Policy language and associated standards
- (U//FOUO) Assured Metadata with integrity verification and reliable binding to source object
- (U//FOUO) Availability of enterprise situation, environment, and IT Component data with integrity verification features
- (U//FOUO) Enterprise Management information regarding domain Certification & Accreditation and its associated configuration, risks, and threat levels
- (U//FOUO) Dynamic Policy Management to push access control policy updates to distributed RAdAC decision points
- (U//FOUO) Requester identity and associated Strength of Mechanism data
- (U//FOUO) Assured user profiles for storing user-based access control heuristics
- (U//FOUO) Discovery process interface for RAdAC to decide about service subscriptions (authorization to use a service) and service disclosure (authorization to know about a service's existence)

2.2.3.1.1.7 (U) Alternatives

(U//FOUO) Attribute-based Access Control (ABAC) offers a more dynamic access control environment than traditional hard-coded access control models since it is based on attribute-value pairs. Because of its similarity to RAdAC with respect to attribute-based inputs, this approach offers a significant advantage in the near term while the harder technical problems of risk determination can be matured through research and development. ABAC can leverage advances in object metadata and enterprise data (both in the form of attribute-value pairs) and can be used as a prototype to address some aspects of operational need determination without requiring the implementation of security risk determination.

(U//FOUO) In ABAC, the digital access control policy would be simpler than in RAdAC since it is essentially rules about required attribute-value pairs for access to a Soft Object, but it does offer dynamic update capabilities through its typical directory-based structure. This approach can also be paired with the complementary Digital Rights Management technology (potentially implemented as additional lists of attribute-value pairs) to address object life-cycle needs. In the long run, this approach will not meet the GIG capabilities required to fully implement the need-to-share enterprise, but it can be used as an alternative technology during early increments.

(U//FOUO) Content-Based Information Security uses encryption and key management techniques to control access to information objects. This approach addresses security risk during the decision to present an access "key" to a given user based on his or her clearance, formal access approvals, and need-to-know. The technological burden in this approach is in the key management rather than on security risk determination or dynamic policy.

2.2.3.1.1.8 (U) Complementary techniques

(U//FOUO) Digital Rights Management, an access control and usage control technology that uses a combination of metadata-based capabilities, cryptographic techniques, and key management. The xRML proposed standard offers significant capability to express digital rights for objects as a set of well-defined attributes.

2.2.3.1.1.9 (U) References

(U//FOUO) Role-Based Access Control (ANSI INCITS 359-2004)

(U//FOUO) Validated Common Criteria protection profiles for access control, including Controlled Access Protection Profile, Labeled Security Protection Profile, Role-Based Access Control Protection Profile

(U//FOUO) Multinational Information Sharing Environment Protection Profile v.1.0

(U//FOUO) Security Assertion Markup Language (SAML) v2.0 (W3C standards organization)

(U//FOUO) eXtensible Access Control Markup Language (XACML) v1.0, OASIS standard 6 Feb 2003; a working draft of v2.0 is available

(U//FOUO) DARPA Agent Mark-up Language (DAML)

(U//FOUO) Web Ontology Language (OWL) v2.0

(U//FOUO) Web DAV Access Control Protocol (IETF standards organization)

(U//FOUO) Content Based Information Security

(U//FOUO) XML Rights Markup Language v2.0

(U//FOUO) Attribute Based Access Control research

- (U//FOUO) SPAWAR: http://www.networkassociates.com/us/_tier0/nailabs/_media/documents/atn.pdf
- (U//FOUO) Mitre: http://portal.acm.org/citation.cfm?id=510781#CIT

2.2.3.2 (U) Assured Metadata

(U//FOUO) GIG Policy-Based Access Control as implemented via RAdAC capabilities relies on certain information conveyed as inputs to its control decision in a consistent and known format. A portion of this control decision input is based on the attributes of the information objects or services that are being requested. These object attributes, including IA related information are relayed by Metadata. To ensure integrity of objects and metadata linkage, this metadata is cryptographically bound to the source (information or service object). Metadata also serves a related function, by providing filterable information supporting discovery and advertisement of data or service object availability for access by qualified GIG users.

(U//FOUO) Specific metadata content and labeling for GIG information and service object is dependant on the object's type. For example, a server-stored information (file) object may have a far different set of metadata attributes than a real-time session object. GIG metadata standards will specify and define these required IA attributes per object type relationships by.

(U//FOUO) The IA related technologies and capability investments that will be required to enable the GIG vision of Policy-Based Access Control in the metadata area include: GIG wide language standardization for IA attributes, trusted metadata creation tools, cryptographic binding of metadata to its source object as well as the ability to reflect and convey metadata for GIG services.

2.2.3.2.1 (U) Metadata Language and Standards

(U//FOUO) Supporting the transition from a GIG need-to-know to a need-to-share information exchange paradigm will require reliable and trusted mechanisms to characterize the IA aspects of information or service objects requested by GIG entities. To provide a reliable supporting mechanism to the GIG Access Control Decision Point process, metadata language/usage must be standardized regarding syntax, semantics, and ontology of IA related information. This standardization provides both the owner (creating organization) and access policy authors with the ability to unambiguously and consistently communicate attributes regarding data about the information or service object, as well as define the attributes of the entities that will support access control decisions for the object instance. This metadata also supports the user information discovery process by providing filterable information content about GIG publicly available objects to authorized users—via GIG search applications.

2.2.3.2.1.1 (U) Technical details

(U//FOUO) GIG Data owners must have the ability to provide granular expression of the value of their information through new fields in the metadata tags. These fields will point to information access policies that define the users, roles, or COIs authorized to access a specific data asset.

(U//FOUO) The IA Component of the GIG will also implement a notion of Quality of Protection (QoP) for data assets. As part of tagging a data asset, a set of security-related properties necessary for protecting the asset would be associated with the asset. Properties can include how to protect the object as it travels across the network, how the data object can be routed, or how the data object must be protected while at rest.

(U//FOUO) The purpose of QoP metadata elements differs from the metadata elements used to describe the contents of an asset. Content-description metadata elements are designed to enable data discovery and sharing. QoP metadata tags define how the data object is to be protected while at rest and in transit. This concept, for instance, will allow the GIG to require routing of highly classified or sensitive information through a more trusted (i.e., better protected) portion of the GIG or require that a user's client support encryption of the information in storage before granting access to the information. Clearly one of the technical issues surrounding these metadata QoP designations are the mechanisms of transformation—especially for transport from metadata to routing request/selection information.

(U//FOUO) Another important aspect when considering metadata usage within the GIG is to consider the types (classes) of objects being requested for access and the potential action context of these object classes. Objects in this context are any information, service, session, application, streaming media, metadata or other resource to which access will be controlled in the GIG. Objects are described as being active or passive with respect to the access control decision process. An object is considered active if it is the cause of the access control decision (i.e., an active object is one that is requesting access to some other object/entity). An object is considered passive if it is the entity that will be shared as a result of the access control decision (i.e., a passive object is the one that is being requested by some other object/entity). There are many classes of objects that will exist in the GIG and be involved in access control decisions. Some possible classes include:

- (U//FOUO) Information objects include any data file, report, document, photograph, database element, or similar types of data object. It might also include metadata that describes other objects. Information objects are arguably the core objects as they typically are what is being shared. They represent all the information that will be resident in the GIG or made available to the GIG from information originators/creators (e.g., Intelligence Community). They are usually passive (that which is being acted upon), and thus their IA attributes often define the minimal requirements for access to the object.

- (U//FOUO) Service objects are executable applications that provide some function for the GIG. They are the services in a service-oriented architecture. Service objects can be both active and passive objects of an access control decision. Services will provide portals to useful information and computational resources on the GIG. People will need to be granted access to services to use them and thus services can also be the passive object of an access control decision. In addition, services can be expected to make independent requests of other services and information objects, or may make requests on behalf of people. When making requests on behalf of a person, services might be expected to provide their own IA attributes to the access control decision process along with those of the person. When independently accessing other services (e.g., service to service interactions), service objects are active objects in the access control decision process.

- (U//FOUO) Session objects are objects that are created as a result of a real-time collaboration between two or more people. A telephone call, a video teleconference, or an online virtual meeting, are examples of collaborative sessions that produce session objects. Session objects are in essence a representation of the collaborative session. They have attributes that describe key characteristics of the session. Session objects will generally be passive objects in an access control decision, and thus the IA attributes of the session will be used to grant or deny access to the session. There may be cases where a session object is also an active object as it might request content be added to the session, such as a data file (e.g., PowerPoint presentation).

- (U//FOUO) Real-time objects are a special class of information objects. Examples of real-time objects are live streaming video and voice, as well as real-time network management/control traffic exchanges. What makes real-time objects special is the temporal aspect of the objects (saving samples to disk turns real-time objects into normal information objects, i.e., these real-time objects are not retained to persistent storage media). Attributes that describe real-time objects must be assigned a priori and thus must be generalized to what the real-time object is expected to be. For IA attributes, this means that the security relevant features of the streaming information must be anticipated. Once IA attributes are established, they will live through the duration of the real-time object.

(U//FOUO) Metadata IA attributes are the foundation of making access control decisions in the GIG. There needs to be a universal agreed-upon set of IA attributes across the GIG. These attributes, in effect, provide a vocabulary for describing security actions. Without a common vocabulary, it is quite difficult, if not impossible, to make meaningful decisions about sharing information. Table 2.2-3 shows the minimum set of IA attributes needed to support policy based access control decision-making via the RAdAC information-sharing model, based on the class of the object.

3160 **Table 2.2-3: (U) Minimum Set of IA Attributes for Access Control Decisions**

This Table is (U)	
Category	**IA Attribute Description/Requirement**
Passive object	Identifier: Provide the GIG unique designation for the object
Passive object	Sensitivity Level: Provide a standards-based designation of object classification and perishability timeframe (**include Operational Need Modifier structure)
Passive object	Data Owner Community of Interest: GIG standards-based COI designator for the organization/activity responsible for creation of the object
Passive object	Access Control Information List/Policy (Direct Data or Pointer): GIG Standards-based Pairing of entities that are allowed access to an object (COI, individual, individual w/ Role/Privilege or groups) and the operations the entity is allowed to perform (read, write, execute, etc.) on the requested object. (**include Operational Need Modifier structure)
Passive object	Time to Live: Length of time an object can be used before it is destroyed automatically by the system as part of an automated life cycle management capability
Passive object	Originator: GIG unique and authenticated identifier linked to the person, organization, or entity that created the object
Passive object	Releaseability: Standards-based designator of countries or GIG external organizations with whom the object may be shared (**include Operational Need Modifier structure)
Passive object	Sanitization Supported: Identifies if real-time sanitization of the object is supported.
Passive object	Security Policy Index: GIG standards-based policy language specifies the various procedures for the object with flexibility/structure to include access protection policy (entity authentication, platform, environment and operational factor scoring) and QoP (**include Operational Need Modifier structure)
Passive object	QoP object life cycle attributes (view only, printable, no-forward, destroy after view, digital rights, etc.) (**include Operational Need Modifier structure)
Passive object	Location: GIG Standards-based designation of virtual path to the object's storage location
Passive object	Timestamp: Time/date information when the object was created or copied.
Passive object	Integrity mechanism: Insure that unauthorized changes to the information object and its IA attributes can be detected
Passive object	Cryptobinding: Cryptographic binding and metadata (supporting access control decision making) to the source object. (Supports prevention of direct access to object w/o metadata based access control decision processing)
Passive object	Split or IA capable filtering of Metadata: Support for both discovery and access control processes
Passive object	Classification/releasability of descriptive metadata itself (not the source object)
Session object	Member IA Attributes: GIG Standards-based listing (pointers) of mandatory privilege/identity IA attribute and value pairings
Session object	Access Control List: List of GIG unique identifier for people allowed to join session paired with GIG unique identifier for approval authority
Session object	Security Level: GIG standards-based parameter indicating how the security level of the session is to be controlled (fixed/float)

This Table is (U)	
Category	**IA Attribute Description/Requirement**
Session object	Session Archive Control: GIG standards-based parameters indicating archive/recording and classification marking required
Session object	Owner/Moderator ID: GIG unique identifier of session owner/moderator
Session object	Session Members: GIG unique identifier of current/past session members
Session object	Session Identifier: Standards-based unique identifier for the session.
Service object	For Access Requests coming from a service object (acting as proxy for the source entity) this structure must address GIG unique ID of service object, as well as GIG unique ID of requesting source *EDITOR'S NOTE: REMAINING SPECIFIC IA ATTRIBUTES FOR SERVICE OBJECT TYPES ARE CURRENTLY UNDER INVESTIGATION*
Real-time object	*EDITOR'S NOTE: SPECIFIC IA ATTRIBUTES FOR REAL-TIME OBJECT TYPES ARE CURRENTLY UNDER INVESTIGATION*
This Table is (U)	

(U//FOUO) **The RAdAC model describes an approach to access control whereby operational necessity can override security risk. In this context, IA attributes might have 'modifiers' in addition to values. Specifically, each designated IA Attribute might have a modifier that describes which, if any, exceptions/overrides to normal policy might be permitted relative to that attribute. Thus, when an access control process is making a decision whether to permit or deny access and encounters a mismatch on a particular IA Attribute, it may use the modifiers in an effort to reach a decision that supports sharing.

2.2.3.2.1.2 (U) Usage considerations

(U//FOUO) The successful usage of a standardized metadata language supporting access control decisions will require a clearly defined and consistently implemented set of IA Attributes and supporting infrastructure/tools capabilities. This set of IA related attributes (labels); their syntax, semantics, and taxonomy form a critical link in the GIG automated access control and discovery processes. The usage and meaning of these IA Attributes must be understood and/or supported via user assisting infrastructure especially for the roles of information owner, access control policy author, and access privilege (operation override) authority. Incorrect usage of these IA Attributes (labels) could result inability to discover or access information by GIG users with the correct operation need and clearance. On the other side of the scale, incorrect IA Attribute usage could result in unintended or unauthorized disclosure of information to a compromised GIG user or service entity.

(U//FOUO) Currently there are two known standards bodies working within the GIG to define metadata language principles for use by their communities. The primary purpose of each group's products are different, and neither standard provides the entire IA Attribute suite needed to support the Policy-Based Access Control Enabler as envisioned in the RAdAC model (See Table 2.2-3 for detailed analysis). However, the Core Enterprise Services (CES) Metadata Working Group, now led by DISA, is attempting to ensure commonality between itself and the IC Metadata Working Group (see attribute comparison Table 2.2-4). Further, discussions have been initiated with both standards groups to investigate and integrate the required IA Attribute and supporting language semantics/syntax into these implementation documents and infrastructures.

Table 2.2-4: (U) IC and CES Metadata Working Groups Attribute Comparison

This Table is (U)

Core Layer Category Set	DDMS Attributes	IC MSP Attributes
The Security elements enable the description of security classification and related fields	Security: Classification Dissemination Controls Releasable To	Security: Classification Dissemination Controls Releasable To
Resource elements enable the descriptors of maintenance and administration information	Title Identifier Creator Publisher Contributor Date Rights Language Type Source	Title Identifier List AuthorInfo Publisher Co-authorInfo Date Rights Language IntelType Source
The Summary Content elements enable the description of concepts and topics	Subject Geospatial Coverage Temporal Coverage Virtual Coverage Description	Subject Geospatial Temporal Virtual Description
The Format elements enable the description of physical attributes to the asset	Format	Media Format

This Table is (U)

(U//FOUO) The IC Metadata Working Group has developed an XML-based standard and schema that supports containers for security marking as prescribed by the CAPCO standard. IC MSP is an implementation of the World Wide Web Consortium's (W3C) specification of the Extensible Markup Language (XML). It consists of a set of XML attributes that may be used to associate security-related metadata with XML elements in documents, web service transactions, or data streams. It is distributed as both an XML entity set and a W3C XML Schema (WXS) so that the XML attributes defined in the standard can be incorporated into any XML document type definition (DTD) or schema. The IC ISM entity set and WXS are controlled vocabularies of terms that are used as the sources for the values of the IC ISM attributes. The IC MSP schemas incorporate the classification and controls attributes defined by the IC Metadata Standard for Information Security Markings (IC ISM). The IC ISM provides the IC with a standard method for tagging CAPCO authorized markings and abbreviations on XML-based information. The standard provides flexibility for each agency to implement their security policy and granularity with respect to security marking.

(U//FOUO) The DoD's Discovery Metadata Specification (DDMS) and supporting XML schema produced by the Core Enterprise Services (CES) Metadata Working Group defines discovery metadata elements for resources posted to community and organizational shared spaces. “Discovery” is the ability to locate data assets through a consistent and flexible search. The DDMS specifies a set of information fields that are to be used to describe any data or service asset that is made known to the enterprise. This CES document serves as a reference guide by laying a foundation for Discovery Services. The document describes the DDMS elements and their logical groupings. It does not provide an interchange specification or substantive implementation guidance. However, there is a roadmap for the development of implementation guides in line with this and higher-level GIG directive documents (see Figure 2.2-2).

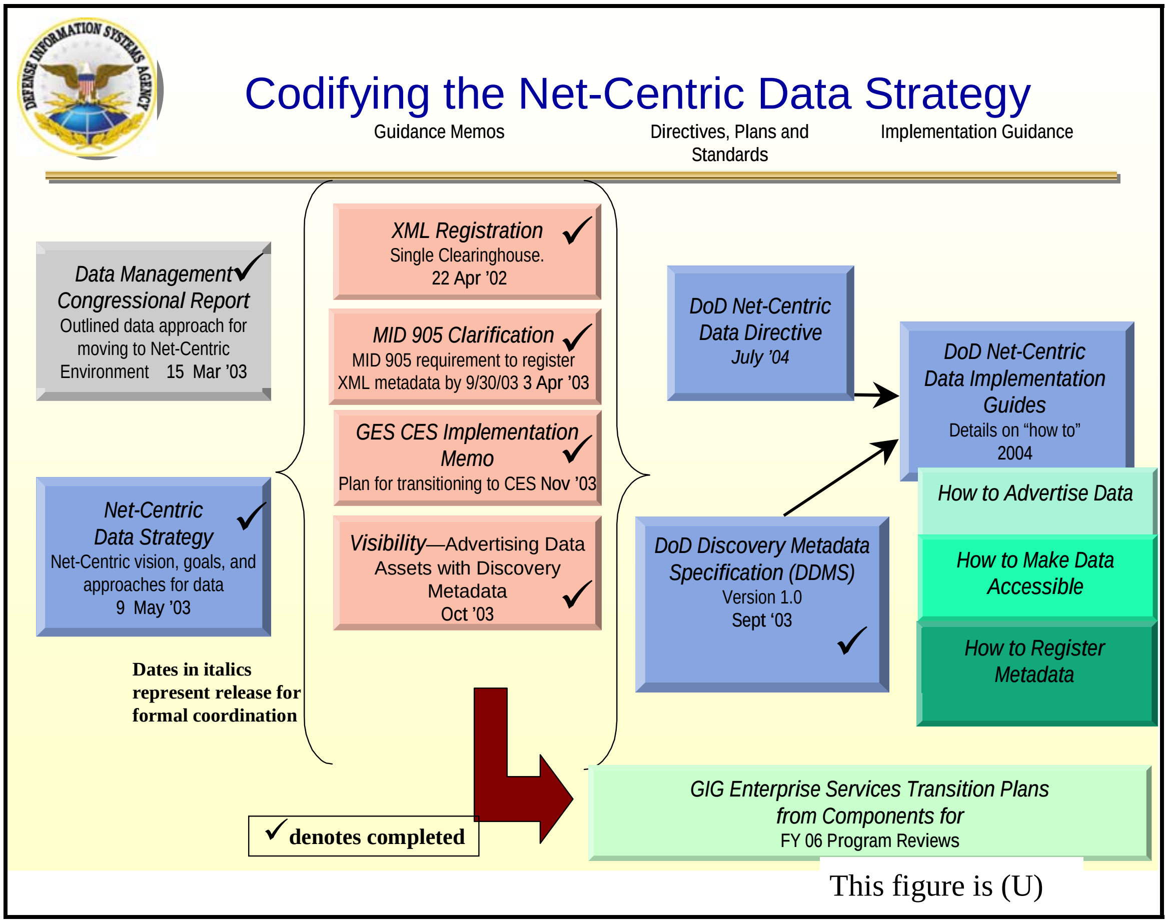


Figure 2.2-2: (U) Codifying the Net-Centric Data Strategy

2.2.3.2.1.2.1 (U) Implementation Issues

(U//FOUO) Some IA metadata attributes sets for information objects may change over time and due to the impact scale. The IA metadata standards/language and supporting infrastructure must support the ability to point (index) to a trusted secondary source for current version attribute information. For instance, if a departmental access policy were hard coded into metadata for all of that department's products, potentially large numbers of information objects must be modified to new hard-coded values if a change policy change occurs over time in this area

(U//FOUO) The metadata language standard must include fields (IA Attributes) within the metadata tag that allow access control decisions to be made on the metadata itself. For example, in some instances, security code words or compartment names are classified themselves

(U//FOUO) It is also paramount, given the critical nature of the metadata tags, that appropriate integrity, data origination and in some cases traffic flow security measures are applied and that the metadata label be securely bound to the object

(U//FOUO) The use of IA attribute modifiers (as described above) will add significant complexity to the IA metadata standards definition.

(U//FOUO) IA metadata attributes will be needed to support both GIG Access Control and Discovery processes. If implementation decisions drive segregation of IA Attributes to differing location (virtual or physical) synchronization of new or changed IA attributes must be addressed

(U//FOUO) Ontology of metadata (referring to input factors for RAdAC computation) is extremely important so that computation logic correctly assesses the risk and the operational need (e.g., is this a Navy “Captain” endorsing operational need or an Air Force “Captain”)

(U//FOUO) It is unclear what implementation method can support the transport-related Quality of Protection (QoP) IA metadata attributes into the transport infrastructure to support routing decisions. For the data at rest portion of IA QoP attributes, commercial-based Digital Right Management capability may provide acceptable and compatible methods give further investigation.

2.2.3.2.1.2.2 (U) Advantages

(U//FOUO) Supports GIG need-to-share vision though discovery process and movement away from “determine at the time of creation” access control lists. (Creator of information may not know who has need of information produced)

(U//FOUO) Supports finer granularity in access control decision making logic

(U//FOUO) Support policy based vs. hard coded, access control decision making that enables rapid changes in GIG situational and environmental factors as well as operational need

2.2.3.2.1.2.3 (U) Risks/Threats/Attacks

(U//FOUO) Attempts to access information object directly on server location by passing metadata/RAdAC Access Control processes

(U//FOUO) Confidentiality of some portions of metadata itself

(U//FOUO) Discovery DOS attacks

(U//FOUO) Access DOS attacks

(U//FOUO) Metadata tags that include compromised identity of the original source of the information and of any entities (e.g., processes) that have modified it prior to posting in its current form

(U//FOUO) Compromised metadata is presented to discovery users (e.g., metadata is maliciously hidden, out of date metadata is maliciously presented)

2.2.3.2.1.3 (U) Maturity

(U//FOUO) As described above, the two GIG standards organizations (CES Metadata Working Group and IC Metadata Working Group) are in the process of defining metadata standards and implementation schemas. These standards are being designed for implementation, using mature and tested commercial standards for internet communication including XML and OWL. Further, GIG usage of XML to support metadata is being configuration managed and standardized via the DOD Metadata Registry and Clearing house (http://diides.ncr.disa.mil/mdregHomePage/mdregHome.portal). Therefore, technical readiness level has been assessed in the Early range (2-3).

2.2.3.2.1.4 (U) Standards

Table 2.2-5: (U) Metadata Standards

This Table is (U)

Standard	Description
Department of Defense Discovery Metadata Specification (DDMS) Version 1.1	Defines discovery metadata elements for resources posted to community and organizational shared spaces. "Discovery" is the ability to locate data assets through a consistent and flexible search.
Intelligence Community Metadata Standards for Information Assurance, Information Security Markings Implementation Guide, Release 2.0	An implementation of the World Wide Web Consortium's specification of the Extensible Markup Language (XML). It consists of a set of XML attributes that may be used to associate security-related metadata with XML elements in documents, webservice transactions, or data streams.
Intelligence Community Metadata Standard for Publications, Implementation Guide, Release 2.0	A set of XML document models that may be used to apply metadata to analytical data to produce publications. IC MSP prescribes element models and associated attributes for use in marking up document-style products for posting on Intelink and other domain servers.
Federal Information Processing Standard FIPS PUB 10-4, April, 1995, Countries, Dependencies, Areas of Special Sovereignty, and Their Principal Administrative Divisions	Provides a list of the basic geopolitical entities in the world, together with the principal administrative divisions that comprise each entity.
Extensible Markup Language (XML) 1.0 (Second Edition) W3C Recommendation, 6 October 2000	Describes a class of data objects called XML documents and partially describes the behavior of computer programs which process them.
Web Ontology Language (OWL) Guide Version 1.0, W3C Working Draft 4 November 2002	Provides a language that can be used to describe the classes and relations between them that are inherent in Web documents and applications.

This Table is (U)

2.2.3.2.1.5 (U) Costs/limitations

(U//FOUO) More resources and time will be required to develop, produce, and maintain these IA related metadata attributes than today's basic security markings and MAC/DAC characteristics. This cost can be off set to some degree by the use of automated metadata creation tools

(U//FOUO) Legacy DoD information and service objects that currently exist or will be produced before metadata standards and infrastructure are available may need to be retrofitted with standard IA Attributes to support RAdAC access control and Discovery process

(U//FOUO) The use of trusted metadata type information tagging for real-time and session object types will increase the GIG's transport and network traffic overhead. Performance impacts should also be investigated early in the design process

(U//FOUO) To avoid the need to retrofit metadata for very large quantities of information objects, IA metadata attributes syntax and semantics must remain "stable" or remain backwards compatible.

2.2.3.2.1.6 (U) Dependencies

(U//FOUO) Access Control Policy Language Standards

(U//FOUO) Metadata Creation Tools

(U//FOUO) Identity and Privilege Management Capacities

2.2.3.2.1.7 (U) Alternatives

(U//FOUO) Depending on the final fidelity/functionality and transition sequence of RAdAC, functionality-less IA Attribute could be included in the metadata language and standards. However later additions could result in metadata, large-scale retrofit impacts.

2.2.3.2.1.8 (U) Complementary techniques

(U//FOUO) Digital Rights Management

2.2.3.2.1.9 (U) References

(U//FOUO) Department of Defense Discovery Metadata Specification (DDMS) Version 1.1

(U//FOUO) Intelligence Community Metadata Standards for Information Assurance, Information Security Markings Implementation Guide, Release 2.0

(U//FOUO) Intelligence Community Metadata Standard for Publications, Implementation Guide, Release 2.0

(U//FOUO) Federal Information Processing Standard FIPS PUB 10-4, April, 1995, Countries, Dependencies, Areas of Special Sovereignty, and Their Principal Administrative Divisions.

2.2.3.2.1.10 (U) Technology/Standards Analysis

Table 2.2-6: (U) Metadata Gap Analysis

This Table is (U)

Category	IA Attribute Description/Requirement	Req. Source	Existing Standards Coverage (Y/N) Identifier	Gap Description/Recommendation	Recommendations and/or Remarks
Passive object/MD Creator Entry	Identifier: Provide GIG unique designation for the object	Tiger Team Report 5/26/2004	Y (IC MSP) Y (DDMS)		IC MSP requires a Universal Unique ID, Identifier List, and a Public Document No.
Passive object/MD Creator Entry	Sensitivity Level: Provide a standards based designation of object classification and perishability timeframe (**include Operational Need Modifier structure)	Tiger Team Report 5/26/2004	Y (IC MSP) Y (IC ISM) Y (DDMS)	Recommend DDMS implement (by reference) the IC ISM markings	IC ISM allows all CAPCO classification markings and dissemination constraints including declassification instructions IC MSP employs IC ISM markings on all block object element types and in the descriptive metadata for the source data DDMS only implements DoD 5200.1-R and does not currently express foreign, SCI, or non-standard classification or declassification

This Table is (U)					
Category	**IA Attribute Description/Requirement**	**Req. Source**	**Existing Standards Coverage (Y/N) Identifier**	**Gap Description/Recommendation**	**Recommendations and/or Remarks**
Passive object/MD Creator Entry	Data Owner Community of Interest: GIG standards based COI designator for the organization/activity responsible for creation of the object	Tiger Team Report 5/26/2004	N (IC MSP) N (DDMS)	IC MSP: Make Affiliation a required field and ensure it aligns to GIG COI designator IC MSP and DDMS: Make UserID a required field and ensure it maps to globally unique GIG UserID DDMS: Make Organization a required field	IC MSP allows specification of 1+ POC's information in PersonalProfileGroup, but Affiliation is optional and COI is missing
Passive object/MD Creator Entry	Access Control Information List/Policy (Direct Data or Pointer): GIG Standards-based Pairing of entities that are allowed access to an object (COI, individual, individual w/ Role/Privilege or groups) and the operations the entity is allowed to perform (read, write, execute, etc.) on the requested object. (**include Operational Need Modifier structure)	Tiger Team Report 5/26/2004	N	See comments/questions	For Access Requests coming from a service object (acting as proxy for the source entity), this structure must address GIG unique ID of service object, as well as GIG unique ID of requesting source
Passive object/MD, Creator Entry	Time to Live: Length of time an object can be used before it is destroyed automatically by the system as part of an automated life cycle management capability	Tiger Team Report 5/26/2004	Y (IC MSP) Y (DDMS)		Supports information cutoff and information "death" dates in the DateList element (IC MSP) and Date element (DDMS)

This Table is (U)					
Category	**IA Attribute Description/Requirement**	**Req. Source**	**Existing Standards Coverage (Y/N) Identifier**	**Gap Description/Recommendation**	**Recommendations and/or Remarks**
Passive object/MD, Creator Entry	Originator: GIG unique and authenticated identifier linked to the person, organization, or entity that created the object	Tiger Team Report 5/26/2004	N (IC MSP) N (DDMS)	IC MSP: Make Affiliation a required field and ensure it aligns to GIG COI designator IC MSP and DDMS: Make UserID a required field and ensure it maps to globally unique GIG UserID DDMS: Make Organization a required field	IC MSP allows specification of 1+ POC's information in PersonalProfileGroup, but Affiliation is optional and COI is missing
Passive object/MD, Creator Entry	Releaseability: Standards-based designator of countries or GIG external organizations with whom the object may be shared (**include Operational Need Modifier structure)	Tiger Team Report 5/26/2004	Y (IC MSP) Y (IC ISM) Y (DDMS)		Support CAPCO and DoD 5200.1-R compliant releasability markings
Passive object/MD, Creator Entry	Sanitization Supported: Identifies if real-time sanitization of the object is supported.	Tiger Team Report 5/26/2004	N (IC MSP) N (DDMS)	Add optional element containing URI to metadata for alternate source or acceptable sanitization service. The URI to alternate metadata should contain a security classification.	

This Table is (U)					
Category	**IA Attribute Description/Requirement**	**Req. Source**	**Existing Standards Coverage (Y/N) Identifier**	**Gap Description/Recommendation**	**Recommendations and/or Remarks**
Passive object/MD, Creator Entry/View	Security Policy Index: GIG standards based policy language specifies the various procedures for the object w/ flexibility/structure to include access protection policy (entity authentication, platform, environment and operational factor scoring) (**include Operational Need Modifier structure)	Tiger Team Report 5/26/2004	N (IC MSP) N (DDMS)	Add mandatory element that indexes the organization security policy that governs access to the information. Index can take the form of a URI or a UUID. Intent is that though the policy may change over time, this reference to it won't need to.	
Passive object/MD, Creator Entry/View	Object lifecycle attributes (view only, printable, no-forward, destroy after view, etc.) (**include Operational Need Modifier structure)	Tiger Team Report 5/26/2004	N (IC MSP) N (DDMS)	Need to add Digital Rights Management (or equivalent attribute fidelity) capability to specify read, modify, forward, copy, destroy, print types of constraints	There's a primitive structure in place for rights management in the Rights element, but it only supports copyright and Privacy Act flags
Passive object/MD, Creator Entry/View	Location: GIG Standards-based designation of virtual path to the object's storage location	Tiger Team Report 5/26/2004	N (IC MSP) N (DDMS)	Add SourceURI field to AdministrativeMetadata element	
Passive object/MD, Creator View	Timestamp: Time/date information when the object was created or copied.	Tiger Team Report 5/26/2004	Y (IC MSP) Y (DDMS)		IC MSP: DateList element contains DatePosted, DatePublished, DateReviewed, DateRevised fields DDMS: Date element contains DateCreated

This Table is (U)					
Category	**IA Attribute Description/Requirement**	**Req. Source**	**Existing Standards Coverage (Y/N) Identifier**	**Gap Description/Recommendation**	**Recommendations and/or Remarks**
Passive object/MD, Creator No Entry/View	Integrity mechanism: Insure that unauthorized changes to the information object and its IA attributes can be detected	Tiger Team Report 5/26/2004	N (IC MSP) N (DDMS)	Append an Integrity element with MetadataIntegrity field and SourceIntegrity field that include name/type/URI of integrity mechanism used	
Passive object/ Infrastructure	Cryptobinding: Cryptographic binding and metadata (supporting access control decision making) to the source object. (Supports prevention of direct access to object w/o metadata based access control decision processing)	GIG IA Arch Docs	N (IC MSP) N (DDMS)	Add a Security element with crypto algorithm designator/URI and a portion of the key needed to access the source	
Passive object/ Infrastructure	Split or IA capable filtering of Metadata: Support for both discovery and access control processes	GIG IA Arch Docs	Y (IC MSP) Y (DDMS)		IC MSP splits Administrative Metadata from Descriptive Metadata DDMS is designed to enable discovery and access control filtering on a single "metacard"
Passive object/ Infrastructure	Classification/releasability of descriptive metadata itself (not the source object)	GIG IA Arch Docs	N (IC MSP) N (DDMS)	Descriptive Metadata only describes the security classification of the source object	
Session object/Owner Entry/View	Member IA Attributes: GIG Standards based listing (pointers) of mandatory privilege/identity IA attribute and value pairings	Tiger Team Report 5/26/2004	N (IC MSP) N (DDMS)	Need to add Element structure that specifies the common qualities of People who can participate in the session	

This Table is (U)					
Category	**IA Attribute Description/Requirement**	**Req. Source**	**Existing Standards Coverage (Y/N) Identifier**	**Gap Description/Recommendation**	**Recommendations and/or Remarks**
Session object/Owner Entry/View	Access Control List: List of GIG unique identifier for people allowed to join session paired with GIG unique identifier for approval authority	Tiger Team Report 5/26/2004	N (IC MSP) N (DDMS)		
Session object/Owner Entry/View	Security Level: GIG standards based parameter indicating how the security level of the session is to be controlled (fixed/float)	Tiger Team Report 5/26/2004	N (IC MSP) N (DDMS)	Add an binary "Fixed" field in the Security element of session DescriptiveMetadata	
Session object/Owner Entry/View	Session Archive Control: GIG standards-based parameters indicating archive/recording and classification marking required	Tiger Team Report 5/26/2004	N (IC MSP) N (DDMS)	Need to add archive/recording fields (Y/N) and URI for archive/recording	Covers classification markings only
Session object/Owner Entry/View	Owner/Moderator ID: GIG unique identifier of owner/moderator of the session	Tiger Team Report 5/26/2004	N (IC MSP) N (DDMS)	Can be adapted using Elements from the PersonalProfileGroup	Assumption: This person is responsible for granting access to the session and is responsible for allowing information objects to be shared in the session
Session object/Owner /View	Session Members: GIG unique identifier of current/past session members	Tiger Team Report 5/26/2004	Y (IC MSP) Y (DDMS)		UUID should suffice for this purpose
Session object/Owner /View	Session Identifier: Standards based unique identifier for the session.	Tiger Team Report 5/26/2004	Y (IC MSP) Y (DDMS)		UUID should suffice for this purpose
This Table is (U)					

2.2.3.2.2 (U) Trusted Metadata Creation Tools

(U//FOUO) The IA metadata attributes are a key element of access control decisions that are at the heart of assured information sharing. Given the pivotal role of these attributes, the policies and supporting creation tools/infrastructure used to generate them can be leveraged to help encourage—or even enforce—the appropriate level of data sharing across the enterprise.

(U//FOUO) It is envisioned that automated process/tools can be developed to support the business processes of the GIG community and can translate these business processes into sharing policies that assist in the application of IA metadata attributes for both sharing and required information security. While such a robust translation capability is beyond the ability of current technologies, the general notion of turning business processes and natural language statements about organization's processes into a machine-readable metadata, supporting policy, tools and infrastructure is supported by current technology—the issue is one of robustness and sophistication. If such a robust capability can be created, it will allow automated processes to facilitate appropriate levels of sharing and security by assisting in the creation of the object metadata IA Attributes.

(U//FOUO) It should be noted that IA Attributes will form only a portion of the overall metadata for GIG information objects. However, due to the critical nature of these elements, a significant amount of complexity and added processing interfaces will be needed to support this metadata subset.

2.2.3.2.2.1 (U) Technical details

(U//FOUO) The following listing provides a brief inventory of capabilities and interfaces that will be required of trusted metadata creation tools and IA attributes for the GIG.

- (U//FOUO) Identity and Privilege management interface: Ensure that the entity (user/process) is authenticated and has the correct privilege to create/validate this metadata for the data owning organization.
- (U//FOUO) Object Identifier CM interface: Assign GIG unique object identifier
- (U//FOUO) Access Control Policy Interface: Allow user to link the correct access control policy or access control list (based on information owner organization’s business rules) as well as directive/pointers to related transport QoP and life cycle QoP policy
- (U//FOUO) Operational Need Entry supporting structure (IA attributes ‘modifiers’ in addition to values. IA attributes might have a modifier that describes which, if any, exceptions to normal policy might be permitted relative to that attribute)
- (U//FOUO) Metadata Integrity Mechanism Interface: Ensure unauthorized changes to metadata are detected
- (U//FOUO) Discovery metadata filtering structure/policy, allows portions of metadata to be filtered from search results unless the user possess required clearance level

- (U//FOUO) Cryptographic Binding Interface: Supports trusted binding of the metadata to the source information object when metadata has been successfully created (syntax check complete)

- (U//FOUO) Trusted transport interface, required for assure pull and push of information related to metadata creation process

(U//FOUO) The following listing provides an inventory of capabilities may be common to the overall metadata creation process (non IA attribute unique)

- (U//FOUO) Context Sensitive User Help Capabilities

- (U//FOUO) Syntax Checker Capability (Note: This may be present for standard metadata requirements; the unique IA attribute will likely add significant code size and interface complexity)

2.2.3.2.2.2 (U) Usage considerations

(U//FOUO) With the advent of XML-based metadata that supports web document publishing and search application, creation tools and templates have been developed to assist users and document owners with the generation/maintenance of supporting metadata. From the prospective of commercial standards, most of these supporting tools are based on the Dublin Core Initiative.

(U//FOUO) The Dublin Core Metadata Initiative is an open forum engaged in the development of interoperable online metadata standards to support a broad range of purposes and business models. The Dublin Core supports standard schemas in both XML and RDFS. Customized metadata creation and maintenance tools—based on the Dublin Core schema—are then developed that reflect the required metadata purpose and business model/policies. These metadata creation/maintenance tools are designed and implemented either by the information owning organization or through customization of commercially available products.

(U//FOUO) However, the IA metadata attributes will require additional capability to ensure trust, security, and unique GIG environment requirement are met for both discovery and access control processes. These IA-related characteristics of metadata support/generation tools were defined in section 2.2.3.2.2.1.

2.2.3.2.2.2.1 (U) Implementation Issues

(U//FOUO) Metadata creation tools may have to support GIG minimum standards related to both discovery and access control as well as providing the user with information related to specific organizational policy. As discussed above, the criticality of the IA Attributes form an access control prospective and will probably make these tools complex. Finally, these tools must be widely distributed, available to a user in a timely manner, be intuitive to a human in their use, and support greater levels of automation during final program timeframes.

2.2.3.2.2.2.2 (U) Advantages

(U//FOUO) Metadata creation tools and supporting infrastructure provide the user or organization entity responsible for creation of data improvements with accuracy and aid with population of the correct metadata information (especially IA Attribute) vs. manual (template only) methods.

2.2.3.2.2.2.3 (U) Risks/Threats/Attacks

(U//FOUO) Unauthorized user “checks in” malicious file/metadata to info storage

(U//FOUO) Unauthorized user attempt to change IA attribute of metadata to gain information object access

(U//FOUO) Authorized user changes metadata

(U//FOUO) Metadata creation tool DOS Attack

(U//FOUO) Compromised metadata creation tool source software

2.2.3.2.2.3 (U) Maturity

(U//FOUO) Clearly, there have been successful implementations and commercial products that provide metadata creation tools based on the Dublin Core metadata standard. As such, the overall technology would receive a TRL score of 7-8. However, the IA related capabilities and interfaces as defined in section 2.2.3.2.2.1 are new, complex and unique to this GIG implementation. Further, one of the key required predecessors needed is a stable metadata standard for IA Attributes. As discussed in the Metadata Language and Standards section, this activity is in the early stages of technology development. Therefore, we assess the overall TRL score for this technology in the Early range (1-2).

2.2.3.2.2.4 (U) Standards

Table 2.2-7: (U) Metadata Tool Standards

This Table is (U)	
Standard	**Description**
Department of Defense Discovery Metadata Specification (DDMS) Version 1.1	Defines discovery metadata elements for resources posted to community and organizational shared spaces. “Discovery” is the ability to locate data assets through a consistent and flexible search.
Intelligence Community Metadata Standards for Information Assurance, Information Security Markings Implementation Guide, Release 2.0	An implementation of the World Wide Web Consortium’s specification of theExtensible Markup Language (XML). It consists of a set of XML attributes that may be used to associate security-related metadata with XML elements in documents, webservice transactions, or data streams.
Intelligence Community Metadata Standard for Publications, Implementation Guide, Release 2.0	A set of XML document models that may be used to apply metadata to analytical data to produce publications. IC MSP prescribes element models and associated attributes for use in marking up document-style products for posting on Intelink and other domain servers
Dublin Core Metadata For Resource Discovery, (RFC 2413 IETF)	Defines interoperable metadata standards and specialized metadata vocabularies for describing resources that enable more intelligent

This Table is (U)	
Standard	**Description**
	information discovery systems.
This Table is (U)	

2.2.3.2.2.5 (U) Costs/limitations

2.2.3.2.2.6 (U) Dependencies

- (U) Access Control Policy Service
- (U) GIG Information Object Identity Assignment Service
- (U) Identity and Privilege Service
- (U) Cryptographic Binding Service

2.2.3.2.2.7 (U) Alternatives

(U//FOUO) Manual (template-based metadata entry) forms with limited syntax checking and external interfaces may be sufficient for the early stages of Dynamic Access (RAdAC based) Control

2.2.3.2.2.8 (U) References

(U//FOUO) Department of Defense Discovery Metadata Specification (DDMS) Version 1.1

(U//FOUO) Intelligence Community Metadata Standards for Information Assurance, Information Security Markings Implementation Guide, Release 2.0

(U//FOUO) Intelligence Community Metadata Standard for Publications, Implementation Guide, Release 2.0

(U) Dublin Core Metadata For Resource Discovery, RFC 2413 IETF

2.2.3.2.3 (U) Crypto-Binding of Metadata to Source Information Object

(U//FOUO) Cryptographically, binding the metadata describing an information object to its source object provides a critical access control integrity mechanism. Crypto-binding ensures at the time of creation or authorized modification that a trusted linkage is established between the two components of an information object (source info and metadata). This capability becomes important to GIG's implementation of Policy Based Access Control via RAdAC because metadata is one of the primary, determining information inputs for access control decisions. Without crypto-binding, the metadata could be altered or maliciously pointed to an invalid metadata tag in order to gain unauthorized access to a source information element.

2.2.3.2.3.1 (U) Technical details

(U//FOUO) The following list provides a brief inventory of capabilities and interfaces that will be required of crypto-binding of metadata to its source information object for the GIG.

- (U//FOUO) Interface capability to GIG metadata creation tools/services
- (U//FOUO) Interfaces to accept and process key and/or digital signature information (as required)
- (U//FOUO) Provide up to type 1 assurance of binding and digest functions for metadata and its source information object
- (U//FOUO) Ability support for rapid decryption of digest (hash file) and return original component files upon receipt of properly authorized command

2.2.3.2.3.2 (U) Usage considerations

(U//FOUO) Research to date indicates that the best standards technology available today to meet the required capabilities of the Cryptographic Binding function are best implemented via the Cryptographic Message Syntax, (RFC 2630) standard. The Cryptographic Message Syntax (CMS) was derived from PKCS #7 version 1.5 as specified in RFC 2315 [PKCS#7].

(U//FOUO) CMS is a data protection encapsulation syntax that employs ASN.1 [X.208-88, X.209-88]. This syntax is used to digitally sign, digest, authenticate, or encrypt arbitrary messages. It supports digital signatures, message authentication codes, and encryption. The syntax allows multiple encapsulations, so one encapsulation envelope can be nested inside another. This capability aligns well with the needs defined for the cryptographic binding functionality (see Figure 2.2-3 Encapsulation Notional diagram).

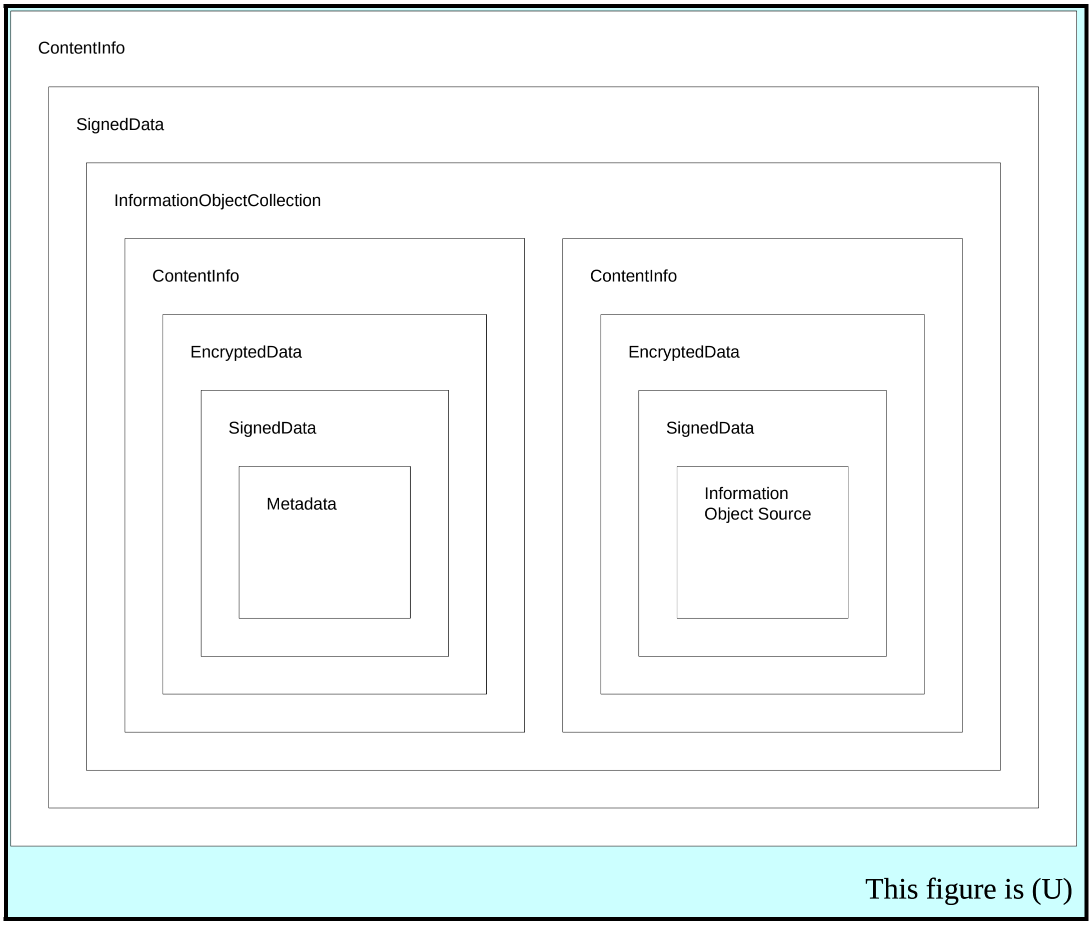


Figure 2.2-3: (U) Encapsulation Notional Diagram

(U//FOUO) CMS implementations must include the SHA-1 message digest algorithm (defined in FIPS Pub 180-10). CMS implementations should include the MD5 message digest algorithm (defined in RFC 1321) as well. CMS implementations must include DSA signature algorithm (defined in FIPS Pub 186). CMS implementations may also include RSA signature algorithm (defined in RFC 2347 for use with SHA-1 and MD5).

2.2.3.2.3.2.1 (U) Implementation Issues

(U//FOUO) The decision as to whether this crypto-binding and decrypt function is a central GIG service or a local plug in to affected applications may affect overall performance, network overhead, and user perception

2.2.3.2.3.2.2 (U) Advantages

(U//FOUO) CMS is flexible and nesting levels are expandable to meet program needs

(U//FOUO) CMS has been successfully implemented in commercial and government network environments

(U//FOUO) CMS provides flexibility to program selection of message digest and signature algorithms. Further, as new encryption and signature algorithms the CMS syntax structure can be expanded to accommodate movement in the technology state of the art.

2.2.3.2.3.2.3 (U) Risks/Threats/Attacks

(U//FOUO) Decryption Analysis/Attack

(U//FOUO) Compromised digital signatures

2.2.3.2.3.3 (U) Maturity

(U//FOUO) Elements of the CMS have a successful lineage from PKCS#7 and a wide variety of successful implementation examples in both commercial and DoD environments for the base encryption, binding, and linkage function. However, the interfaces to other GIG applications/services and potential distributed nature of this function will drive a small to moderate level of new development. As such, we judge the overall TRL level of this technology to be in the Early to Emerging range (3-4).

2.2.3.2.3.4 (U) Standards

Table 2.2-8: (U) Standards on Cryptographic Binding

This Table is (U)	
Standard	**Description**
Cryptographic Message Syntax, IETF (RFC 2630)	This syntax is used to digitally sign, digest, authenticate, or encrypt arbitrary messages.
PKCS #7 version 1.5 9 IETF (RFC 2315)	Describes a general syntax for data that may have cryptography applied to it, such as digital signatures and digital envelopes. The syntax admits recursion. It also allows arbitrary attributes, such as signing time, to be authenticated along with the content of a message, and provides for other attributes such as countersignatures to be associated with a signature.
SHA-1 (FIPS Pub 180-10)	Standard specifies a Secure Hash Algorithm, SHA-1, for computing a condensed representation of a message or a data file
MD5 IETF (RFC 1321)	Standard describes the MD5 message-digest algorithm. The algorithm takes as input a message of arbitrary length and produces as output a 128-bit "fingerprint" or "message digest" of the input.
Hashed Message Authentication Codes (FIPS PUB 198)	Standard describes a keyed-hash message authentication code (HMAC), a mechanism for message authentication using cryptographic hash functions. HMAC can be used with any iterative Approved cryptographic hash function, in combination with a shared secret key.
This Table is (U)	

2.2.3.2.3.5 (U) Dependencies

(U) Key management infrastructure

(U) Metadata standards/infrastructure

2.2.3.2.3.6 (U) Alternatives

(U//FOUO) SHA-1 in concert with RSA signature service could be implemented and used without standardized syntax (CMS). Syntax relation processing and infrastructure would need to be maintained in entirety by DoD/GIG.

2.2.3.2.3.7 (U) Complementary techniques

(U) Described in section 2.2.3.2.3.2.

2.2.3.2.3.8 (U) References

(U) Cryptographic Message Syntax, IETF (RFC 2630)

(U) PKCS #7 version 1.5 9 IETF (RFC 2315)

(U) SHA-1 (FIPS Pub 180-10)

(U) MD5 IETF (RFC 1321)

(U) RSA IETF (RFC 2347)

(U) Hashed Message Authentication Codes (RFC 2401)

2.2.3.3 (U) Digital Access Control Policy

(U//FOUO) Influencing all aspects of the RAdAC model is the digital access control policy (DACP). It serves as an input to the Core RAdAC functions and as the deciding factor for allowing or denying access. Although RAdAC will need specific capabilities in its DACP, these policy needs should fold into the larger GIG dynamic policy effort. Some potential technologies being examined for that enabler are WS-Policy, Standard Deontic Logic, and artificial intelligence constructs. The scope of this section addresses only the RAdAC-specific needs for DACP and assumes that the dynamic policy enabler provides the necessary distributed functionality (e.g., secure update, revocation, currency validation, and caching for off-line use).

(U//FOUO) The RAdAC model depicts DACP as influencing all aspects of internal RAdAC behavior. In this role, DACP must be expressive enough to address the following:

- (U//FOUO) Minimum number of required inputs to calculate risk and operational need
- (U//FOUO) Relative weighting of the various inputs for risk and operational need
- (U//FOUO) Relative weighting of risk versus operational need for the final decision
- (U//FOUO) Ability to express stateful access control rules (e.g., successive failed access attempts)
- (U//FOUO) Ability to express policy according to enterprise and COI roles
- (U//FOUO) Ability to negotiate two or more conflicting access control rules
- (U//FOUO) Ability to negotiate access control policy with neighboring security domains

in order to define an access control boundary interface that is agreeable to both sides

- (U//FOUO) Ability to express and automatically select between multiple policies based on nationality or security domain
- (U//FOUO) Ability to express more granular or more restrictive access control policies at each successive echelon down the chain of command
- (U//FOUO) Ability to dynamically tighten or loosen access control policy based on situation (INFOCON, proximity to enemy forces, etc.).
- (U//FOUO) Ability to reach a decision deterministically within bounded time

(U//FOUO) DACP also requires expressiveness to support RAdAC output. For example, the policy engine may recognize a specific request as having a compelling operational need but having too risky an IT Component to release the information to. In this case, policy should be expressive enough to conclude that an alternate path (alternate Course of Action, or COA) for this LIMFAC should be examined. For this role, DACP expressiveness must address:

- (U//FOUO) Ability to understand and specify in human- and machine-readable terms the limiting factors (LIMFACs) that contributed to a failed access attempt
- (U//FOUO) Ability to reason whether an alternate COA could sway the decision (e.g., an uncleared user attempting to access Top Secret information could never be allowed—regardless of the QoP offered by a specific route—because of national policy).
- (U//FOUO) Integrity and timestamp features to avert malicious attacks
- (U//FOUO) Ability to select and reason about various enterprise alternatives (e.g., alternate routing for higher QoP, imposed digital rights to limit risk, automatic sanitization options, nearby neighbors with sufficient access) via comparison and "what if" scenarios

(U//FOUO) Finally, extraordinary operations support requires that DACP be able to handle policy exceptions that are able to authorize normally disallowed actions due to an extremely urgent operational need. Most likely, such an authorization would be tightly constrained by time controls (very limited access period) and additional access/distribution controls (very minimal set of well-defined actions) to limit risk.

2.2.3.3.1.1 (U) Technical details

(U//FOUO) DACP must be able to reach a decision based on risk computation, operational need computation, and policy input. Final decision logic uses digital policy to compare risk and need computations against acceptable thresholds, specify a decision, and generate a corresponding access token of some sort, generate a decision rationale, and generate an audit record.

(U//FOUO) The DACP language features must support conflict detection and resolution, negotiation across RAdAC domains/COIs, dynamic update, ontology specification, and human readability. Policy must be able to be securely updated, revoked, and enforced within acceptable performance margins to ensure currency with dynamic enterprise policy.

(U//FOUO) RAdAC must have a grammar that can succinctly express decision rationale that is unequivocally tied to the input received, including the ability to list limiting factors (LIMFACs) in both a machine-understandable and human-readable format.

(U//FOUO) While the ability to discover and select an alternate COA is a highly desirable feature of a RAdAC-enabled system, embedment of this capability within the core RAdAC model would severely impact the performance of the access decision process. Rather than embed this functionality within RAdAC, the preferred approach is to have an offload capability to a separate service to perform this analysis and make recommendations. Similar digital access control policy can be used by this ACOA service to reason about alternatives it considers, and this ACOA service may optionally provide a user interface for the User to select between [possibly less desirable] alternate COAs.

(U//FOUO) The ability to handle temporal exceptions for extraordinary operations via dispensations or work flows is a critical DACP feature to enable RAdAC dynamic operations support. Certain deontic languages provide this capability in the form of “dispensations” that augment the DACP based on a compelling temporal need. Other approaches include work flows to address the specific LIMFACs identified in access control decisions. Regardless of the technical approach, great care must be taken to constrain where dispensations are allowed and not allowed within the policy language due to national law or immutable operational policy. For example, dispensations may be allowed for dissemination of a classified document to a cleared User, without formal access approval, given compelling operational need but may never be allowed for an uncleared User. Dispensations may be the most appropriate way for digital policy to annotate and reason about a commander or supervisor’s consent for a User’s operational need to know a particular piece of information.

(U//FOUO) The policy must be robust enough offer a low error rate (i.e., meet extremely stringent false negative and false positive rates). Since RAdAC would be replacing the traditional Mandatory Access Control model objectively, false positives in particular cannot be tolerated for risk of information disclosure. Dispensations for exception handling must be constrained in such a way that guarantees select portions of digital access control policy will comply with national law.

2.2.3.3.1.2 (U)·Usage considerations

(U//FOUO) Since DACP forms the primary underpinning of the RAdAC model, its implementation will require significant analysis and community vetting. It will also require protections against a wide range of security threats since it will be a likely target of IW attack.

2.2.3.3.1.2.1 (U) Implementation Issues

(U//FOUO) Conflicting laws and policies - Established laws and organization security policies require sufficient clearance, formal access approval, and need to know to establish authorization for classified information. We need to do an assessment of how RAdAC maps to these requirements (e.g., does operational need equate to need to know) to determine which laws and organization policies require amendment

(U//FOUO) Human understandable access control policy - Enterprise managers and certifiers will want a human-readable format to the Access Control Policy to examine and evaluate its specifications, but RAdAC will need fast machine-readable versions of the same policy to meet performance needs

(U//FOUO) Supporting decision rationale - A format/grammar must be developed to express the rationale for an access control decision and any associated LIMFACs and deciding factors. This grammar may need to be purposefully limited, though, to avoid disclosing too much information about the current DACP and how to influence its decisions

(U//FOUO) Minimal acceptable input parameters - Need to do research in defining the minimum quorum and pedigree of input parameters necessary to make an access decision with bounded risk. Does this minimal set vary based on the Environment (CONUS versus tactical) or Situation (exercise versus active engagement)? Are heuristics employed only if the access is not decidable given the other input parameters, or is it always part of the decision process?

(U//FOUO) IT Component integration - DACP and RAdAC's decision output must be tightly integrated with the policies that affect the management of the IT Components. This avoids situations where RAdAC allows access through a given enterprise route but then the enterprise routes the information over a different path because of other decision metrics. Digital rights policy enforcement must be tightly integrated with the end user equipment portion of IT Components so that the rights embedded with the information object are strictly enforced

2.2.3.3.1.2.2 (U) Advantages

(U//FOUO) Supports dynamic operations through update and reasoning about operational need and security risk for access control decisions

(U//FOUO) Facilitates expression in human understandable format for analysis and update

(U//FOUO) Supports exception handling for extraordinary operations with compelling operational need

(U//FOUO) Extends beyond the access decision to address soft object life cycle and distribution controls

2.2.3.3.1.2.3 (U) Risks/Threats/Attacks

(U//FOUO) Spoofing or man-in-the-middle unauthorized modification of policy updates

(U//FOUO) Replay of access control requests or decisions to cause a denial of service

(U//FOUO) Unintentional misconfiguration of DACP can introduce access denial or confidentiality breaches

(U//FOUO) Exception handling could potentially be misused by insiders to gain access to unauthorized soft objects (e.g., exaggerating operational need)

2.2.3.3.1.3 (U) Technology/Standards Analysis

(U) Specific technologies and standards for digital policy are analyzed in Section 2.4. This subsection applies that analysis specifically to the digital policy needs for Policy-Based Access Control.

(U//FOUO) XML Access Control Markup Language (XACML) has been pushed within the web services and DoD network-centric initiatives and has reached significant maturity as a result, but it has some serious limitations for digital access control policy. The largest limitations are its present inability to understand ontology and to resolve conflicting policy assertions. A third limitation is in the area of dispensations, since they can only be approximated through a policy update and policy revocation after a specified period.

(U//FOUO) The standards being developed under the W3C semantic web initiative appear to meet the wide range of needs for digital access control policy. They address ontology (via OWL) and use deontic logic to capture, reason through, and apply business rules according to underlying mathematics. Certain deontic logic technologies such as Rei and KaOS offer the ability to create and apply dynamic dispensation rules as well. Though the expressiveness of the standards appear sufficient to cover the needs for digital access control policy, further analysis needs to be done to extend the deontic logic math model to address specific access control needs, verify performance of the technologies, and verify scalability to an enterprise level.

2.2.4 (U) Distributed Policy Based Access Control: Gap Analysis

2.2.4.1 (U) Core RAdAC: Gap Analysis

(U//FOUO) The Core RAdAC functions are in their infancy with respect to concept formulation, standards development, and technology implementation, as shown from a summary level in Table 2.2-9. Industry really will not benefit from RAdAC as the Government will, so it is not surprising to see little research and development in this area. Industry is showing interest in role-based access control and now attribute-based access control, but RAdAC's unique features put it on a complementary but dissimilar technology path.

(U//FOUO) The following technology gaps exist for RAdAC:

- (U//FOUO) <u>Attribute Based Access Control standard</u> - Although there is research and even initial product offerings for ABAC-based products, there is no IETF or Government standard. Cisco and Maxware have proprietary products, and Network Associates is doing research funded by SPAWAR, but none meets all of the attribute requirements for RAdAC. Since we are looking at ABAC as an interim implementation of RAdAC, we could employ a proprietary solution while RAdAC is being explored and developed in parallel. But we would do so at the potential risk of it becoming the GIG standard if RAdAC is not realizable for a presently unknown technical or political reason. Prudence dictates that we have an alternate fallback standard in place, given the current immaturity of RAdAC and its critical role in the enterprise.

- (U//FOUO) Protection Profiles - There are no current or planned protection profiles that address RAdAC or attribute-based access control. Existing protection profiles are limited to Orange Book approximations. These protection profiles are necessary to establish the minimum security protections required for any implementation of RAdAC.

- (U//FOUO) RAdAC standard - Since industry is not moving in the RAdAC direction, there are no formal representations of architecture, interface definitions, performance requirements, or protocol requirements.

- (U//FOUO) RAdAC math model - RAdAC needs an underlying math model to meet medium and high assurance implementation requirements and to assist in the transformation from a DAC and MAC access control culture. This math model needs to include the digital access control policy since the two are so tightly integrated. Further extensions to the deontic logic math model need to be accomplished to apply it specifically to the access control domain and prove mappings of certain policy constructs to traditional DAC and MAC access control models.

- (U//FOUO) Input parameter ontology - All attributes that feed the RAdAC model need to have an ontology that is accessible and standardized. This applies to attributes of IT Components, Environment, Situation, Soft Objects (metadata), and People.

Table 2.2-9: (U) Technology Adequacy for Access Control

This Table is (U)

		CoreAccess Control	Digital Rights	Access Control Policy	Required Capability (RCD attribute)
Enabler Attributes	Risk & Need Determination		N/A		IAAC4
	Math model		N/A		IAAC4
	Decision logic		N/A		IAAC1, IAAC4, IAAC7
	Ontology	N/A	N/A		IAAC4
	Exception handling		N/A		IAAC5
	Conflict resolution		N/A		IAAC7
	Object Lifecycle				IAAC8
	Protection Profile				IAAC9

This Table is (U)

2.2.4.2 (U) Assured Metadata: Gap Analysis

(U//FOUO) From an overall prospective, as shown in Table 2.2-10: (U) Technology Adequacy for Metadata, the technology and functionality gaps in the assured metadata area will not require the same levels of technology leaps, or major innovations in comparison to the RAdAC portion of this enabler or other technologies needed in the GIG.

(U//FOUO) For the metadata standards area, both the IC and DoD are working on the definition of standards to support discovery and marking of information that will be part of the GIG. Both groups have built their standards implementation/schemas based on a widely proven and available commercial language/technology (XML and OWL). Further, an initial gap analysis has been completed which compares the capabilities (IA attributes) needed to support RAdAC style access control decision making and discovery (See Table 2.2-6: (U) Metadata Gap Analysis) with these standards. The process of coordinating between these two organizations has been started to ensure that these required IA attributes are integrated into these implementation standards. Stability or backward compatibility of these IA attributes from a syntax and semantics prospective will be critical. If not well planned, changes after final approval will likely ripple to changes in supporting tools and infrastructure, or could affect large quantities of previously populated information object metadata records. Finally, prior to stabilizing the metadata standards and IA attributes, it is strongly recommended that further studies be conducted examining the impact and potential for optimization regarding the increased of metadata IA granularity and its potential GIG impact to network traffic/overhead, especially for real-time and session object types.

(U//FOUO) The development of trusted metadata creation tools can parallel the metadata standards in initial design. However, final development, integration, and testing will be dependant on a stable and accepted metadata standard(s) with required IA attributes. There have been successful implementations and commercial products that provide metadata creation tools based on the XML web publishing, Dublin Core metadata standard, and have been applied to their communities' metadata standard and creation needs in the commercial environment. However, the IA related capabilities and interfaces as defined in section 2.2.3.2.2.1 are new, complex, and unique to this GIG implementation.

(U//FOUO) In the area of cryptographic binding of metadata to its source information, Cryptographic Message Syntax (RFC2630) is the recommended technology standard. This syntax standard provides the capability to support selectable digest and signature algorithms. It is also expandable to support the potential inclusion of other algorithms/standards as technology progresses. However, like the metadata creation tools, the GIG and IA interface aspects required of this capability will remain a technical challenge.

(U//FOUO) NOTE: The metadata IA attributes analysis is currently focused on the various forms of GIG information objects. IA metadata attributes unique to service objects and their supporting tools are currently in work and will be addressed in the next release of the technology roadmap.

Table 2.2-10: (U) Technology Adequacy for Metadata

This Table is (U)

		Metadata Standards/ Language	Metadata Creation Tools	Metadata Cryptographic Binding	Required Capability (RCD attribute)
Enabler Attributes	Commercial Standards Based				
	GIG or IA assurance (unique) interfaces provided				IAIL1, IAIL2, IAIL3, IAIL4, IAIL6, IAIL13, IAIL16
	GIG Governance (Standards) Bodies in Place			N/A	IAIL5, IAIL10, IAIL12, IAIL16, IAIL17
	Need to Share/Control Granularity supported			N/A	IAIL9, IAIL10, IAIL12, IAIL14
	RAdAC value and Modifier Construct supported			N/A	IAIL9
	IA Attribute Internal Consistency (syntax Checking)			N/A	IAIL9, IAIL10, IAIL20
	Cryptographic Performance up to Type 1 Assurance	N/A	N/A		IAIL15

This Table is (U)

2.2.4.3 (U) Digital Access Control Policy: Gap Analysis

(U//FOUO) The proposed OWL v2.0 standard for ontology and the deontic language implementations Rei and KaOS appear to meet the expressiveness required for digital access control policy, but there is significant work needed to realize a complete implementation that will meet GIG information-sharing requirements. The following list describes the major gaps.

- (U) DACP standard. A digital access control policy standard that uses ontology and deontic languages needs to be developed based on the underlying math model. This standard will address the access control policy grammar, exception handling, business rules about allowable and disallowable policy constructs, and business rules for policy negotiation and deconfliction.

- (U) Digital Rights Management integration specification. Digital Rights can be viewed as a static projection of digital access control policy onto a particular soft object. There is currently ongoing research in the Digital Rights realm and proposed standards, but none of them specify a relationship to digital access control policy. An analysis of their relationships, digital rights implementation (XrML or otherwise), and Policy Enforcement Point interface is necessary to complete the end-to-end access control of GIG information and support the transition to a “need-to-share” culture.

2.2.5 (U) Policy Based Access Control: Recommendations and Timelines

(U//FOUO) The following is a list of prioritized distributed policy-based access control gap closure recommendations or actions. They are listed from highest to lowest priority.

- (U//FOUO) Develop Attribute-Based Access Control standard
- (U//FOUO) Develop ABAC and RAdAC Protection Profiles
- (U//FOUO) Develop RAdAC standard
- (U//FOUO) Develop RAdAC math model
- (U//FOUO) Conduct RAdAC prototyping for requirements discovery. This activity feeds input ontology development, RAdAC standard development, DACP standard development, and Digital Rights integration specification
- (U//FOUO) Work with IC and CES Metadata working groups to integrated IA attributes into a standard in accordance with detailed analysis, or (preferred) support the merge of these standards, and ensure IA RAdAC required attributes are included
- (U//FOUO) Begin early design of metadata creation tools in parallel with metadata standards definition to ensure IA specific attributes and authorization interface needs are addressed
- (U//FOUO) Develop input parameter ontology
- (U//FOUO) Conduct study on RAdAC performance and optimization techniques
- (U//FOUO) Conduct RAdAC pilot program to test fielding and operational issues
- (U//FOUO) Develop DACP standard with associated business rules
- (U//FOUO) Develop Digital Rights integration specification
- (U//FOUO) Conduct study on impact and potential for optimization of metadata IA granularity related to GIG network traffic/overhead
- (U//FOUO) Continue work of defining the GIG services metadata tagging capabilities potential technologies

(U//FOUO) Figure 2.2-1 summarizes timeframes for these closure recommendations.

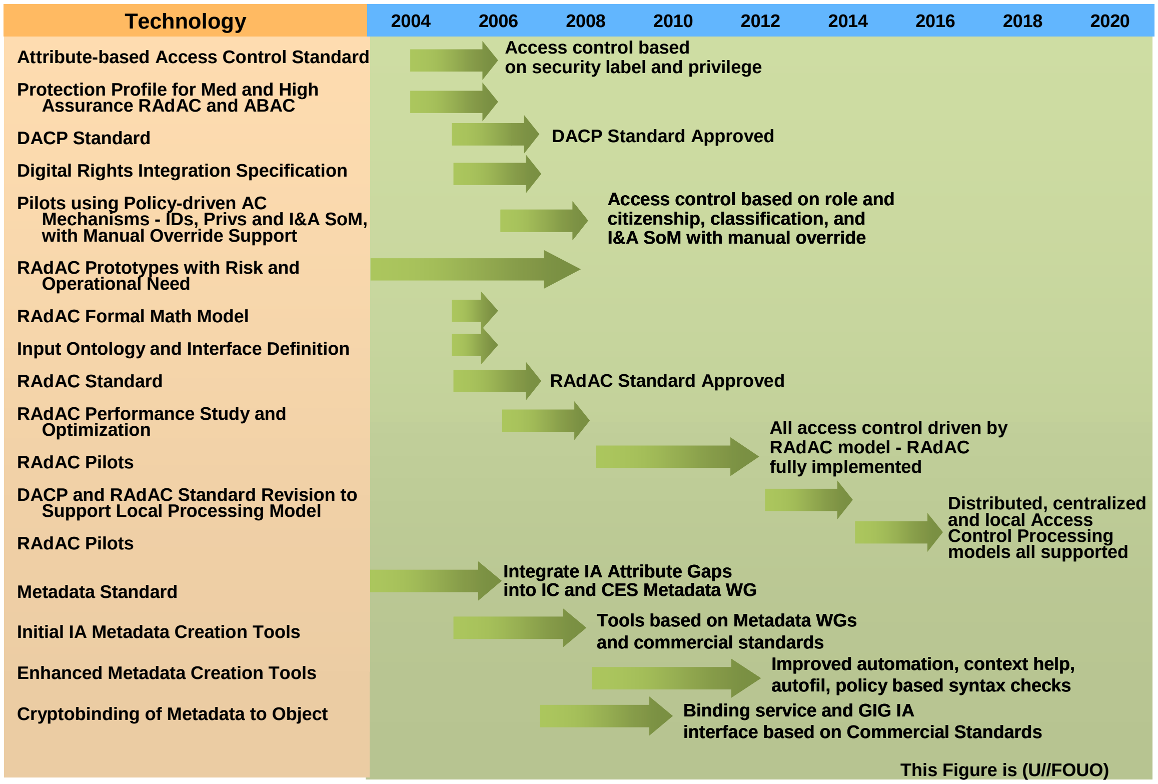


Figure 2.2-4: (U) Policy-Based Access Control Gap Closure Timelines

2.3 (U) PROTECTION OF USER INFORMATION

((U//FOUO) Protection of user information provides the protection of data-at-rest and data-in-transit from end-entity to end-entity. For applications based on the client-server model common to much of today's networks, this GIG vision would provide integrity, confidentiality, and other required security services in both directions between the originating client and the responding server. For peer-to-peer-based applications, this provides those same services between the corresponding peers. For applications-based on other models, appropriate security services will be applied.

(U//FOUO) End-to-end protection of user information does not always mean security services are provided between the true endpoints of communication. There is always a trade-off to be made. For example, if end-to-end confidentiality is provided, that implies that the information is encrypted between the requesting client and the responding server. That means that GIG-provided or organization-provided infrastructure devices such as intrusion detection systems and firewalls cannot examine the data as it passes. This makes it difficult to detect and stop malicious code such as viruses or worms, it makes it difficult to perform content-based filtering (e.g., Spam checking), and it makes it more difficult to detect and stop intrusions. In this scenario, the client node itself must provide all security. This may not be feasible for commercial operating systems and products—even in the 2020 time frame—and it may make it very difficult to detect attacks from authorized GIG insiders.

(U//FOUO) Even within single devices, end-to-end protection of user information may have different meanings depending on the specific application or organization. For example, multiple users or user identifiers may share a single end-point (e.g., multiple users may share a client node, and multiple services may share a single server). End-to-end communications security in this context may mean client-to-server security or it may mean end–user-to-server-identifier security.

(U//FOUO) Thus, depending on the enterprise and U.S. Government policy, different applications may have end-to-end security between clients and servers or communicating peers; or they may have end-to-end security between organizational enclaves; or between other points. These situations are entirely consistent with the GIG Vision.

(U//FOUO) However, there is much work to be done before this vision can be accomplished. The current environment includes many systems operating in different domains and at different security levels. Communication and interoperation among these domains and across these different security levels is not always possible. True end-to-end secure communications cannot be provided in the current or near-term GIG.

(U//FOUO) For the current and near-term GIG implementations, Cross-Domain Solutions provides the necessary secure interoperation. Applications and communications must be secured within a single security level—within a domain. Then, interactions between domains are allowed by using cross-domain solutions (e.g., guards, gateways and firewalls, and specific routing techniques).

(U//FOUO) As the GIG evolves from the current capability set to the vision system, this will gradually change. As core systems are fielded that can allow the merger of domains and supporting multiple classification levels in a system, less emphasis will be placed on such cross-domain solutions as guards and content filters. More emphasis will be placed on security at the end-points, whether those end-points are enclave boundaries or client nodes themselves.

2.3.1 (U) GIG Benefits Due to Protection of User Information

(U//FOUO) The Information Assurance constructs used to support Protection of User Information provide the following services to the GIG:

- (U//FOUO) Protects information in accordance with enterprise-wide policy and the data owner's specified Quality of Protection (QoP)
- (U//FOUO) Allows multiple users to use a single workstation so a user can walk up to a client and access their information
- (U//FOUO) Allows access to multiple levels of information on the same platform without compromising that information (i.e., trusted hardware/software platforms)
- (U//FOUO) Protects against the analysis of network protocol information, traffic volume, and covert channels
- (U//FOUO) Provides user-to-user protection of secure voice traffic from speaker to listener.

2.3.2 (U) Protection of User Information: Description

(U//FOUO) Protection of user information provides the protection of data objects at rest and the protection of data-in-transit. Data-at-rest protection is the protection of data objects while they are stored in repositories across the GIG and within a client's local environment. Data-in-transit protection is the protection of information flows as they move across the GIG within all levels of the transmission protocol stack, including application, network, and link level.

(U//FOUO) Protection of User Information also includes the concept of the GIG Black Core. The Black Core is the packet-based portion of the GIG, where packet level protections are provided between end entities. Over time, end entities providing packet level protections move from the network boundaries to the enclave boundaries to the end clients. Circuits within the GIG will be protected with circuit encryption. In addition, some high risk links may need additional protections if the risk of traffic analysis or other threat is exceptionally high. Possible solutions include link encryption and TRANSEC.

(U//FOUO) Classified information will be protected using high assurance (Type 1) mechanisms, while unclassified information will be protected using evaluated commercial mechanisms. To support the end state capability to enable users to access the proper information, encryption boundaries must be able to support both Type 1 and commercial mechanisms. Encryption products must also have access to the proper key material to protect all classifications of information.

(U//FOUO) The protection of user information must support large numbers of dynamic communities of interest (COI). Support for COIs does not necessarily imply encrypted tunnels between COI members. COIs can also be accomplished through other mechanisms, such as filtering (e.g., Access Control Lists [ACLs]), or logical separation (e.g., Multi Protocol Label Switching [MPLS] Labeled Switch Path [LSPs]). Sufficient auditing mechanisms are necessary to track the establishment and termination of COIs.

(U//FOUO) To support connectivity between GIG networks and coalition networks, mechanisms are necessary that allow information flows to pass between coalition partners and the GIG. Each coalition network will be different and require different security mechanisms and procedures. Some coalition networks will be owned and operated by the U.S. with partners using resources. Other will be owned and operated by allies with U.S. users. Still others will be owned and managed by a number of different allies all intended to seamlessly interconnect. These mechanisms are enforced in a construct referred to as a trust manager.

(U//FOUO) Trust managers enforce policy for connections to coalition partners and allow or disallow individual connections between GIG users and coalition partners. Trust managers can filter traffic types, allow or disallow specific users, monitor information flows, or enforce any other policy required for coalition connections.

(U//FOUO) Whenever GIG systems interact with coalition or an ally's resources, both sides of the connection will have security mechanisms in place. While the GIG will be able to control policy on the GIG side of the connection, the coalition partner will set policy for the other side of the connection. This compounds the problem of information sharing with coalition partners. This is similar to the issues with sharing information across GIG systems, as both policies must be coordinated. Information shared with coalition partners could include all types of data objects and data formats, including data files, messaging, video, streaming video, voice, and web traffic.

(U//FOUO) The GIG will require that clients and computing platforms (i.e., hardware and software) have more inherent trust than they do today. Devices directly accessible by users—running a variety of user applications and connected to untrusted networks—tend to be the least trustworthy devices in the network. They are ripe targets for malicious code attacks and mis-configuration. However, the GIG will rely on clients to do a variety of security-critical functions (e.g., maintain domain separation when accessing information at various levels of sensitivity, support authentication of a user to the infrastructure, support authentication of a client to the infrastructure, properly label data, enforce local security policy, properly encrypt data).

(U//FOUO) In today's system high environments (i.e., JWICS, SIPRNet), less trust in clients is required since all users within an environment have an equivalent level of trust. While placing trust in clients today may seem unreasonable, the GIG Vision requires that procedures and mechanisms be in place to allow clients to perform critical security functions. A higher level of trust within clients is especially important as coalition users and networks are connected to the GIG and as today's system high boundaries are eliminated. A higher level of trust is required for all devices in the GIG— not just end user clients. All devices in the GIG will be required to perform security related functions, and there must be a sufficient degree of trust in these devices for them to reasonably execute their functions.

(U//FOUO) The GIG, however, will consist of IT devices (i.e., routers, servers, clients) with varying levels of trust. The GIG will use a concept referred to as Quality of Protection for data objects. As part of the data labeling, an object will be associated with security properties and policies necessary for protecting the object. Properties can include:

- (U//FOUO) How to protect the object as it travels across the network (e.g., commercial grade vs. Type 1 protections, object and/or packet or link level data-in-transit protection requirements)
- (U//FOUO) How the data object can be routed (e.g., must be contained within the GIG, can flow to or through networks external to the GIG, such as coalition networks or the Internet)
- (U//FOUO) How the data object must be protected while at rest. QoP is different from metadata that describes the contents of the object.

(U//FOUO) Metadata is designed to enable discovery and data sharing. QoP defines how a data object is protected while it is at rest and in transit. When QoP is defined, it should not reveal attributes related to the data originator or client. Policy-Based Access Control will provide the enforcement mechanisms to assure the specified QoP is provided.

(U//FOUO) Data-at-rest protection will be required for some types of data (e.g., for extremely sensitive information) and for certain environments (e.g., information stored on a local client within a hostile environment). The requirements for data-at-rest protection will be identified through a protection policy, such as within a data object or client's protection policy. For shared information, data-at-rest protection must be provided at the object level, where an object is defined as a file or pieces of a file, such as paragraphs. This leads to a large range of object types.

(U//FOUO) All data objects must be protected properly. GIG users must be able to discover and access objects. This will require a key management infrastructure that can dynamically deliver the key material to access objects requested by the user. Data-at-rest for local clients can be provided in a number of ways, including media encryption mechanisms.

(U//FOUO) Protection of data-in-transit consists of the ability to provide confidentiality, integrity, and authentication services to information as it is transmitted within the GIG. The QoP information will describe the services needed for any specific data object.

(U//FOUO) Protection of data-in-transit includes providing traffic flow security (TFS). TFS should be provided for all high-risk links in the GIG but could also be provided for medium or low-risk links. In general, TFS protections include mechanisms that protect against network mapping and traffic analysis. In general, the lower in the protocol stack confidentiality is applied, the greater the TFS benefit. For circuits, end-to-end circuit encryption provides traffic flow security. For IP networks a variety of mechanisms can be used. For IP environments where the communications links are circuit based and the routers are protected one option could be hop-by-hop link encryption applied to the communications links to provide traffic flow security for encrypted packet traffic. TFS mechanisms, however, have a performance impact and should be carefully matched against the risk for the information flow.

(U//FOUO) Protection of data-in-transit also includes the ability to prevent unauthorized transmission of data within the GIG. A covert channel is an unauthorized information flow that is precluded by the network's security policies. Covert channels must be eliminated to permit global access of information required within the GIG.

(U//FOUO) Network layer data-in-transit security is the protection of IP packets as they flow across the GIG. Protection could be from enclave to enclave to enclave, or from host to host. High Assurance Internet Protocol Encryptor (HAIPE)-compliant devices will be used to provide Type 1 data-in-transit network layer security for the GIG. At a minimum, Unclassified data will be protected using medium robustness Type 3 solutions.

(U//FOUO) Speech traffic (Voice over IP [VoIP]) within the GIG can be protected at the Network Layer. Currently, HAIPE can only provide enclave-to-enclave protection. In the future, when HAIPE is integrated into end-systems, the protection can be migrated from the enclave level back to the user level. This functionality will require the development of a new mode within the HAIPE standard to meet the real-time performance requirements of a Voice over Secure IP (VoSIP) terminal.

(U//FOUO) Media gateways can also be defined to extend speech capability beyond the GIG to legacy circuit-based systems, although Network Layer security is not effective beyond such a gateway. Therefore, using HAIPE to protect speech traffic would require Red gateways to legacy circuit-switched networks. The appropriate security (e.g. Future Narrow Band Digital Terminal [FNBDT]) would have to be applied on the circuit-switched side of the gateway to protect the speech traffic over a legacy network.

(U//FOUO) Application Layer data-in-transit security is the protection of information as it flows from one end user terminal to another, where the end user terminals apply the protection mechanisms and the protected information is not accessible at any point during transmission.

(U//FOUO) Within the GIG, most speech traffic is carried across circuit switched networks. Speech traffic in circuit switched networks is protected at the application layer using Secure Terminal Unit-Third Generation (STU-III) or Secure Terminal Equipment (STE) products. STU and STE products provide application layer speaker to listener security. Future secure voice products and architectures must consider interoperability with existing secure voice products (e.g., secure voice products used by NATO, tactical secure voice products.)

(U//FOUO) Application layer protection of speech traffic (VoIP) within the GIG could also be accomplished through development of secure VoIP terminals. Interoperability of secure VoIP terminals will require a common implementation of FNBDT over IP. Secure VoIP terminals will provide end-to-end, Multiple Single Levels of security across the Black Core. That is, although only one session is permitted on each end terminal at a time, subsequent sessions can be established at different security levels.

(U//FOUO) Secure VoIP terminals can be placed on the Black Core to provide end-to-end, Application Layer security across the Black Core. VoIP gateways can also be developed to provide interoperability with legacy FNBDT products on the Public Switched Telephone Network (PSTN). Such a gateway requires access to the IP network on one side and access to appropriate circuit-based networks on the other. The gateway then provides interworking between the IP protocol stack and a circuit-based modem. There are some issues (e.g., transcoding in the gateway needs to be disabled), but these issues can be resolved to provide a Black gateway solution for the FNBDT Application Layer security approach.

(U//FOUO) Secure VoIP terminals can also be placed in Red enclaves to provide user-to-user security, whereas HAIPEs fronting the enclaves only provide enclave-to-enclave security. FNBDT can be overlaid on HAIPE to provide this user-to-user level of security.

(U//FOUO) Overlaying FNBDT on top of HAIPE provides several benefits. First, it provides confidentiality of user voice traffic within the enclave. Second, it allows the security level of the voice session to be based on the clearances of the users rather than the security level of the Red enclave. Finally, it enables interoperability between phones attached to networks at different security levels (cross-domain solutions).

(U//FOUO) Communications between two secure VoIP terminals in different enclaves, where the two enclaves are in the same security domain, is relatively straightforward. The HAIPE fronting the two enclaves perform network level encryption between the enclaves, and the Secure VoIP phones attached to the Red enclave networks perform FNBDT application level encryption between the two users. In this scenario, the users are not restricted to a conversation at the security level of the Red enclave networks. For example, two users with Top Secret clearances could hold a Top Secret conversation on phones attached to secret level enclave networks. Note this scenario utilizes Red enclave call control (call control in the security domain of the Red enclaves).

(U//FOUO) From the above examples, it can be seen that there are potentially multiple domains of call control. A single user and associated secure VoIP terminal could potentially use multiple call control domains. The call control domain used for an instance of communications would be based on the security domains of the networks where the local and remote users' secure VoIP terminals are attached.

(U//FOUO) Data-in-transit protection can also be applied to the GIG at protocol stack layers other than Network and Application. This protection may be in place of or in addition to security at other layers. Specifically, many individual links within the GIG may require protection appropriate for the Physical Layer, such as transmission security (TRANSEC). Security at this layer provides protection that cannot be obtained at other layers, including:

- (U//FOUO) Anti-Jam (A/J)
- (U//FOUO) Low Probability of Interception/Detection (LPI/LPD)
- (U//FOUO) Traffic Flow Security (TFS) and Traffic Analysis Protection
- (U//FOUO) Signals Analysis Protection
- (U//FOUO) Protocol and Header Cover/Packet Masking
- (U//FOUO) TRANSEC Isolation for Major Sets of Users

2.3.3 (U) Protection of User Information: Technologies

(U//FOUO) The technologies in this enabler are organized into technologies that provide data-at-rest protection, data-in-transit protection, trusted platforms, trusted applications, Cross Domain Solutions, and non-repudiation. The data-in-transit protection technologies are further organized by protocols layers. Non-repudiation and Cross Domain Solutions are broken out separately because they do not fit cleanly into either data-at-rest or data-in-transit.

2.3.3.1 (U) Technologies for Protecting Data-at-Rest

(U) EDITOR'S NOTE: MATERIAL ON PROTECTING DATA-AT-REST WILL BE ADDED IN A FUTURE RELEASE. SECTIONS ARE PROVIDED BELOW THAT REFLECT THE TYPE OF CONTENT PLANNED.

2.3.3.1.1 (U) Cryptography

(U) There are several applications of cryptography for protecting data at rest including encryption, signing/authentication, binding, and integrity checking. Cryptographic capability may reside in dedicated security devices or be provided within the host itself.

2.3.3.1.1.1 (U) Storage Networks and Networked Storage Operations

(U) There is an increasing trend towards the use of storage networks to share storage resources (data and/or capacity) or to provide geographic distribution of storage assets for increased availability and survivability. Network Attached Storage (NAS) and Storage Area Networks (SAN) are the two primary approaches. SANs introduce or exacerbate security problems due to the following:

- (U) A very large amount of information may be contained within one system
- (U) Storage resources may need to be shared between domains or enclaves
- (U) Storage assets may be directly accessible from the network including the WAN
- (U) The storage network management infrastructure needs protection
- (U) Access enforcement is remote from data owners/producers and data users
- (U) Possible distribution of storage elements over large distances.

(U) A NAS provides file storage using Network File System (NFS) or Common Internet File System (CIFS) over TCP/IP. A SAN provides virtual disk volume storage using a Small Computer Systems Interface (SCSI) family protocol. IP-based storage protocols are being developed and implemented. Elements of a storage network include storage arrays, switches, host bus adapters/hosts, and security devices.

(U) Whether storage networks are used or not, there are also existing storage operations across the GIG networks that have similar security concerns. These include replication of data among distributed sites, distributed data stores, backup and restorable operations between sites, and archives of data to remote sites.

(U//FOUO) In general, security standards and specifications for network storage are less mature than those for communications security. There are no common definitions of security services across vendors. Across security services vendors, common definitions are lacking and a corresponding shortfall in security products and security features for storage devices exist.

2.3.3.1.2 (U) Data Backup & Archive

2.3.3.1.3 (U) Data Destruction

2.3.3.1.4 (U) Labeling

2.3.3.1.5 (U) Periods Processing

2.3.3.1.6 (U) Physical Controls

2.3.3.1.7 (U) Quality of Protection

(U//FOUO) Quality of Protection is a ranked set of end-to-end protection properties of a system that collectively describe how resources will be protected within that system. These properties may include network infrastructure characteristics, client IT characteristics and the cryptographic capabilities of the IT and network components. A resource will not be made available to a user unless the resource protection requirements can be met by the QoP level of the system or another policy supersedes these requirements. The QoP of the system is not typically one fixed level but is instead a range of available capabilities that can be utilized by the component enforcing the resource protection requirements. For example, in routing a packet, a path that meets the packet's resource protection requirements is utilized if available and if in accordance with QoS and other applicable policy. For data-at-rest, the QoP includes such topics as controls for copying the data, moving the data, and printing.

2.3.3.2 (U) Technologies for Protecting Data-in-Transit

2.3.3.2.1 (U) Application Layer Technologies

(U) Application Layer security technologies typically secure primary user data and may also secure aspects of the application protocols themselves. Application Layer security can provide protection of user data while in transit, and in some case while stored. These security technologies do not generally provide protection against traffic analysis, or attacks on lower layer protocols (e.g., IP).

(U) Application security technologies are characteristically different for real-time applications than for non-real-time applications. Real-time applications include technologies such as streaming audio and video. Non-real-time applications include such technologies as email, web browsing and web services.

2.3.3.2.1.1 (U) Non-Real-Time Data Technologies

(U) Three basic classes of technology are used to provide non-real-time application security. These consist of the following:

- (U) Traditional Layered Application Security Technologies
- (U) Session Security Technologies
- (U) Web Services Security Technologies

(U) Security technologies for applications that operate in non real-time apply a wide spectrum of techniques to the problem of securing primary user data end-to-end. Such technologies generally provide a generic framework for using basic security mechanisms—such as cryptography, one-way functions, and security protocols—to potentially provide abstract security services within the context of a particular type of information exchange between cooperating applications. Figure 2.3-1 shows this relationship.

(U) Nearly all non-real-time applications interface to the layer below them in a connection-oriented manner, making the dialog between the applications subject to security concerns. Generally, security will be provided by a sub-layer that operates below the application and applies security mechanisms to the communication. In the Application Layer, such security functionality is usually modeled as a discrete functional object rather than a sub-layer because same security mechanisms might be applied in different ways to different applications, leading to layering inconsistencies. Some security objects are generic, and can offer service to multiple applications. Others are tightly coupled to or embedded in the applications that they serve. Like the applications themselves, security objects exchange protocols with peer objects.

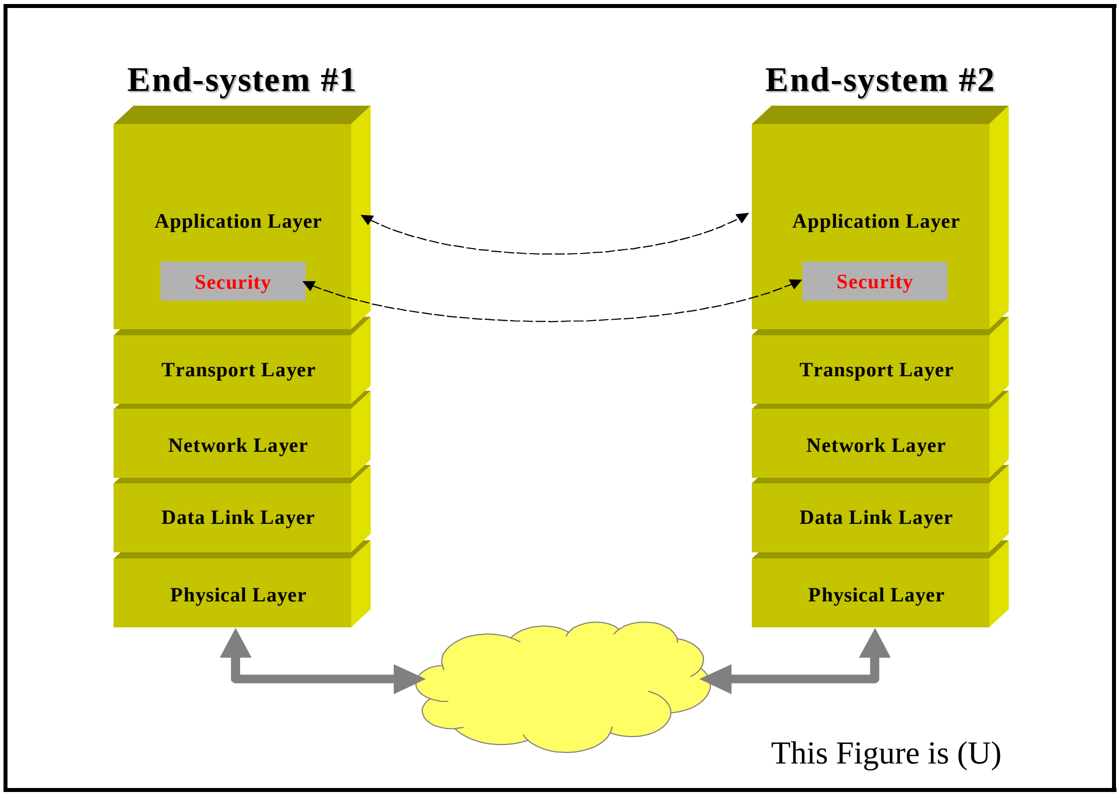


Figure 2.3-1: (U) Context of Non Real-Time Application Security[4]

(U) Application security tailors the application of security techniques to the specific needs of the application. This means that the security object can selectively apply security techniques differently to discrete fields or messages exchanged with the peer. Security mechanisms can be applied selectively to specific fields, using different keys for different fields, to achieve different services. This is superior in many regards, such as better accommodating Cross Domain Solutions (CDS) by selectively leaving parts of the application data readable—or even unprotected—for use by CDS boundary protection devices.

(U) The concept of layered communications entails each layer operating semi-autonomously and adding its own additional protocol wrapper or control information to the data of the layer above it. Figure 2.3-2 illustrates this concept. The layer in question (termed the "n" layer) provides service to the layer above (termed the "n+1" layer since it is one layer higher), and receives service from the layer below (termed the "n-1" layer). Service is provided at the Service Access Point (SAP) for layer "n" also termed the (n)SAP. To request service from the "n" layer, the "n+1" layer conceptually submits a request at the (n)SAP along with an Interface Data Unit (IDU) to support the request. The IDU consists of a Service Data Unit (SDU) (i.e., payload data from the "n+1" layer) and Protocol Control Information (PCI) associated with the requested "n" layer services.

[4] (U) Note that this figure uses the more commonly used OSI terminology for the layers, but omits the Presentation and Session layers as in the Internet model because comparatively few applications in use today employ these layers.

(U) A concrete example of this is an Application Programming Interface, which typically consists of a calling address (analogous to the (n)SAP) and a convention for passing parameters (analogous to the SDU and PCI). The SDU and PCI passed to the “n” layer are used to formulate the SDU that is passed to the “n-1” layer, and thus the virtual Protocol Data Unit (PDU) exchanged with the “n” layer of a communicating peer end-system. Security sub-layers or objects continue to follow this layered communication model. Security objects provide service to the application above, encapsulate the incoming SDU from the “n+1” layer as part of the SDU that is passed down to the “n-1” layer, and incorporate some of the supplied PCI in the SDU and PCI that are passed down.

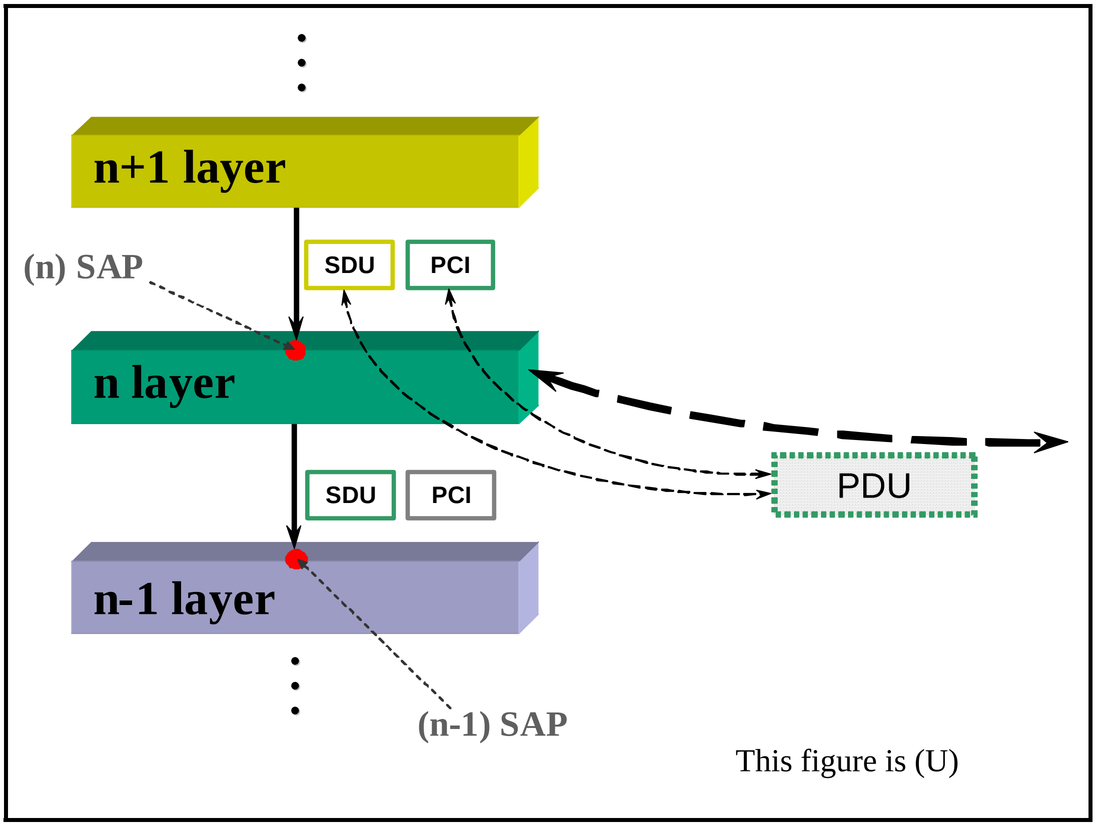


Figure 2.3-2: (U) Layered Protocol Wrapping Concept

(U) Engineering application security entails working through trade-offs among different choices of mechanisms used to provide the desired protection. Application security usually contains embedded use of cryptography, one-way functions, and security protocols. Cryptography is used to render selected portions of the application data unreadable to any entities not possessing the proper key material. One-way functions are a class of mathematical operations that are elementary to perform, but prohibitively difficult to reverse. They are often used to embed irreversibility in application security operations. Security protocols are the backbone of application security. They define the data structures (i.e., what to send) and dialogs (i.e., when to send it) used to exchange information between the application security peer entities. Protocols may resemble a simple one-way exchange or a complex conversation replete with security handshakes. Security protocol design is crucial because most abstract security services (e.g., integrity, authentication) are not possible except in a specific protocol context. Design of the application security usually relies equally on all three of these types of mechanisms as part of an overall open system security solution. Cryptography alone is not enough as a bad security protocol can hamper or compromise good cryptography.

2.3.3.2.1.1.1 (U) Traditional Application Security Technologies

(U) Most development to date of application security has focused on so-called traditional layered technologies. These are characterized by implementation of a standardized security element in the application layer with a strong relationship to and binding with the target application. Such technology has been applied to many applications including message handling or electronic mail, web hypertext, and file transfer. Development of security elements in this manner represents the old school of application security because doing so can require many years of standardization, implementation, and testing to realize workable secure solutions. However, considerable development of traditional application security has already taken place, and it can be leveraged by the GIG.

2.3.3.2.1.1.1.1 (U) Technical Detail

2.3.3.2.1.1.1.1.1 (U) Secure Messaging

(U) Secure messaging is a good example of the evolutionary development of traditional layered application security technology. Early messaging was based on Simple Mail Transfer Protocol (SMTP) and (MSGFMT). It offered ASCII-only messages without attachments, security, or other advanced features. Many implementations of these messaging standards were created including, most notably, the SENDMAIL implementation which was bundled free with most UNIX implementations.

(U) The International Telegraph and Telephone Consultative Committee (CCITT)[5] entered the scene by developing its X.400 series of recommendations. X.400 aimed to provide a full-function messaging system. However, the initial version of the X.400 released in 1984 contained no provision for security features.

[5] (U) CCITT has since reorganized into the International Telecommunications Union (ITU) Telecommunications Standardization Sector (ITU-T).

(U) As the U.S. government was becoming interested in X.400, as part of the developing Open Systems Interconnection (OSI) protocol stack, NSA began development of the Message Security Protocol (MSP) as part of the Secure Data Network System (SDNS). MSP provided security to either X.400 or SMTP through the addition of a connectionless security protocol wrapper around the message content. MSP evolved further as part of the Multilevel Information Systems Security Initiative (MISSI), and was eventually offered to the Allies as Allied Communications Publication (ACP) 120 [CSP].

(U) ACP 120 is used in the presently deployed Defense Message System (DMS). The DMS implementation of ACP 120 works with the FORTEZZA card and the FORTEZZA Certificate Management Infrastructure (CMI) to provide encryption and digital signature for formal military messages. When properly used, ACP 120 is capable of providing the following security services:

- (U) Proof of Content Origin
- (U) Proof of Content Receipt
- (U) Content Confidentiality
- (U) Content Integrity
- (U) Common Security Protocol (CSP) Integrity
- (U) Security Labeling
- (U) Rules-based Access Control
- (U) Secure Mail List Support.

(U) While MSP was being developed and deployed, CCITT was working on their security solution for X.400. This solution is today primarily described in X.400, X.402, and X.411. The X.400 security solution potentially offered all of the same services as CSP, but offered too much flexibility and insufficient definition of necessary embedded security objects to suffice without additional profiling. With the demise of OSI, X.400 security has never achieved widespread implementation or deployment and is no longer a major factor in the evolution of secure messaging.

(U) With the wholesale abandonment of OSI and X.400, emphasis returned to providing security for SMTP and the recently standardized Multipurpose Internet Mail Extensions (MIME). Work began on the Privacy Enhanced Mail ([PEM) project within the IETF.

(U) While PEM ultimately failed[6], it led to the private development of the Public Key Cryptographic Standard (PKCS) #7 PKCS7 and the Secure MIME (S/MIME) development by RSA Data Security Inc. An industry desire to expand the available choices of S/MIME cryptography and achieve compatibility with MSP led to the development of S/MIME v3 in the IETF. The S/MIME working group of the IETF has produced several proposed standards of note, including CMS, MSG, CERT, and ESS. Like MSP, S/MIME v3 provides a wide range of security services including:

- (U) Proof of Content Origin
- (U) Proof of Content Receipt
- (U) Content Confidentiality
- (U) Content Integrity
- (U) S/MIME Protocol Integrity
- (U) Security Labeling
- (U) Secure Mail List Support.

(U) Unlike some application security mechanisms, the specification of the CMS is inherently designed to be a flexible and reusable module in the S/MIME design. It thereby has the potential to support other communications or non-communications applications. This arrangement is illustrated in Figure 2.3-3. This situation already demonstrably exists in that the IETF Pubic Key Infrastructure X.509 (PKIX) working group has used CMS as the foundation for its successful Certificate Management Messages over CMS (CMC) protocol. The IETF Long-Term Archive and Notary Services (LTANS) working group is planning to similarly use CMS as a foundation for their EvidenceRecord format. CMS can (and is) similarly used locally for file encryption outside of the communication stack. The inherent flexibility of this modular style of application security development has the potential to lead to expedited development of traditional layered application security elements in the future.

[6] (U) The failure of PEM had much to do with the conflicting requirements of the changing messaging environment at the time of its development. Interested readers should also see the MIME Object Security Services [MOSS] enhancement of PEM.

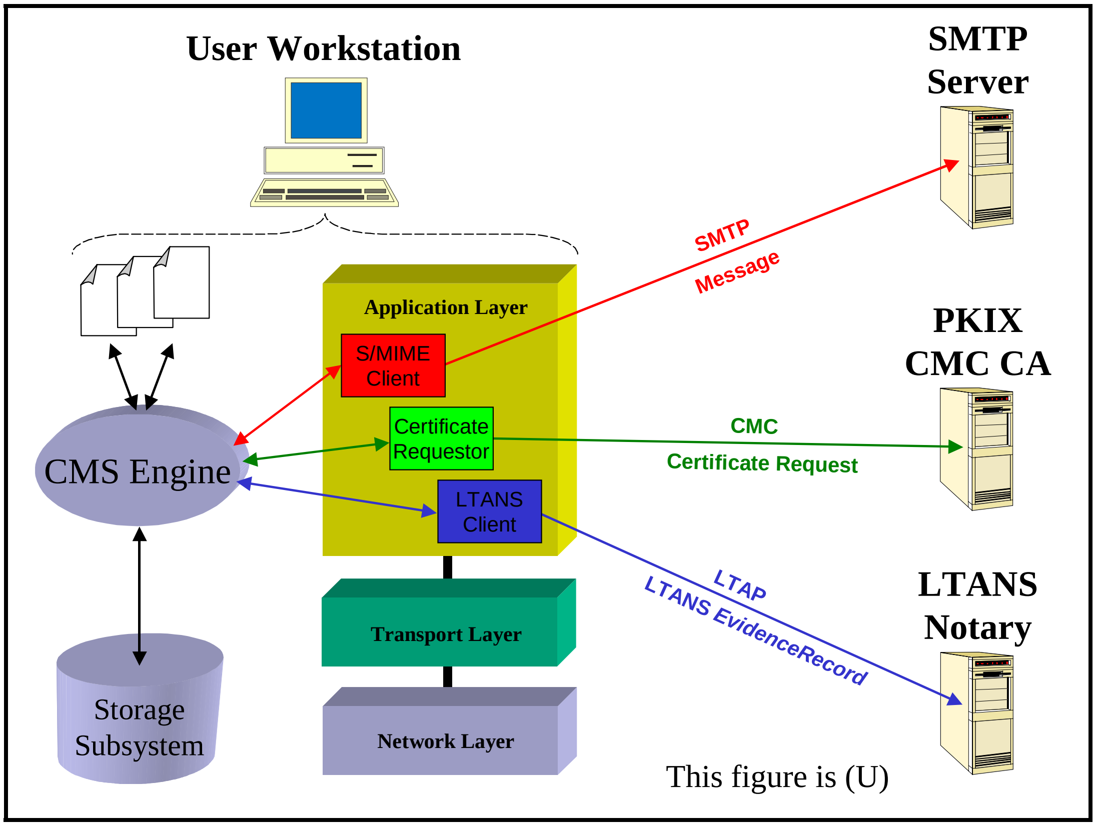


Figure 2.3-3: (U) CMS Supports S/MIME and Other Secure Applications

(U) Another parallel track of secure messaging evolution is that of the Pretty Good Privacy (PGP) development. PGP began as a piece of freeware code for file encryption with public keys. Through the introduction of the PGP-MIME specification, it also began to provide application security for SMTP/MIME messaging. The OpenPGP working group of the IETF is continuing to develop and advance PGP format and PGP-MIME as Internet standards. PGP is not believed to be in use within DoD. While not as capable as S/MIME, OpenPGP nevertheless remains a competitor in the marketplace. OpenPGP is capable of providing the following security services:

- (U) Proof of Content Origin
- (U) Content Confidentiality
- (U) Content Integrity.

(U) The development and evolution of application security for message handling is a long story that is continuing to be written. The widespread use of secure messaging, both in DoD and industry will make it an important factor for the GIG for many years.

2.3.3.2.1.1.1.1.2 (U) Web Security

(U) Traditional layered application security technology has been applied to provide security for web browsing with the HyperText Transfer Protocol (HTTP), but have yielded only limited success. This is not to be confused with Secure Sockets Layer (SSL) which is a different technology covered in Section 2.3.3.2.1.1.2.1.1. Salient examples of application security for web browsing include Secure HTTP (S-HTTP) and the IETF Web Distributed Authoring and Versioning (WebDAV) effort.

(U) The S-HTTP protocol extended the basic HTTP/1.1 protocol to provide mechanisms that can deliver strong authentication, integrity, and confidentiality. While HTTPAuth provided a means for password and digest-based authentication and integrity for HTTP, it failed to provide strong authentication or confidentiality. S-HTTP defines its own URL protocol designator, namely shttp.[7] When a S-HTTP aware client or server detects a shttp URL, it individually secures HTTP requests and responses while preserving the transaction model and implementation characteristics of HTTP. The S-HTTP protocol provides flexibility in choice of cryptographic algorithms, key management mechanisms, and security policy by negotiating each option between the client and server. Key exchange mechanisms include a password-style keying, manually shared secret keys, and public key. The protocol has the capacity to use a variety of cryptographic message formats, including CMS and MOSS. While effective, S-HTTP was never very successful as a technology. S-HTTP is seldom used today.

(U) The IETF WebDAV working group is now taking another look at developing traditional application security for web transactions that are not well served by the simple SSL treatment of the application layer. Web authoring, as opposed to browsing, has a strong emphasis on authentication, access control, and privileges. The Distributed Authoring Protocol (WebDAV) built a framework for distributed authoring by standardizing HTTP extensions to support overwrite prevention (locking), metadata management (properties), and namespace management (copy, move, collections). The Access Control Protocol (WebDAV-AC) builds upon this to provide the means for a web client to read and modify access control lists (ACLs) that instruct a server whether to allow or deny operations upon a resource. As implementation of List Based Access Control (LBAC) fundamentally requires authentication, WebDAV-AC relies on existing authentication mechanisms defined for use with HTTP. WebDAV-AC particularly specifies that if the basic authentication in HTTPAuth is used, it must be performed over secure transport such as TLS. WebDAV is still a relatively young developing standard, and its support level in industry is still relatively low.

(U) On the whole, traditional application security has not been very competitive for web security. The ubiquitous support for SSL in web browsers has made a lot of past web security efforts irrelevant. However, the WebDAV effort appears to recognize the limits of SSL technology, and is exploring richer application security features.

[7] (U) This should not be confused with "https," which signifies SSL/TLS security.

2.3.3.2.1.1.1.1.3 (U) Strong Client Authentication

(U) Several applications are known to employ traditional application security elements as part of their authentication design. Some employ the reusable module philosophy already demonstrated for S/MIME. Applications known to do this include the Simple Authentication and Security Layer (SASL), the Post Office Protocol (POP3), the Internet Message Access Protocol (IMAP), the Application Configuration Access Protocol (ACAP), and security extensions to the File Transfer Protocol (FTP).

(U) Addition of strong client authentication has been a success from a standardization perspective. However, from an implementation and deployment standpoint the track record is spotty. IMAP products commonly incorporate strong authentication. However, POP products still commonly rely on plaintext passwords. ACAP products have been very slow to emerge overall, but incorporate strong security where they exist. FTP products incorporating strong authentication exist, but are seldom used today.

2.3.3.2.1.1.1.1.4 (U) Summary

(U) As their widespread use demonstrates, traditional layered application security technologies have a large footprint in industry and represent a mature, stable development path. However, their maturity is offset by the long lead time associated with their evolution. It is noteworthy, though, that a lot of this lead time is not profitless in that it allows interest and enthusiasm for the standard to build in the vendor community before the standard is finalized. This can lead to improved standards and more widespread support among vendors. Many application security protocols now also embrace a modular design philosophy, such as employed by S/MIME, CMC, and others, which promises to shorten future development cycles.

2.3.3.2.1.1.1.2 (U) Usage Considerations

(U) Application security is generally highly tailored to the needs of the application in question. Since the applications that will make up the GIG are necessarily a moving target, it is difficult to provide a comprehensive overview of specific application security technologies that are of potential interest to the GIG community. That type of analysis is best conducted within the framework of a particular project (e.g., DMS, GDS).

2.3.3.2.1.1.1.3 (U) Maturity

(U) Overall, the traditional application security technology represents a mature foundation for GIG development. Many application security standards have been developed. Some have succeeded while others have failed. Products for most widely-used applications offer at least some form of embedded application security today. Secured application products are generally available, functional, reasonably secure, interoperable and well tested. However, the maturity of specific application security varies dramatically. S/MIME security is widely available in mail clients. Embedded strong IMAP authentication is likewise mature and dependable. Toolkits are available to facilitate rapid integration of many technologies into existing or new product developments. However, other more negative examples, such as strong POP3 authentication and S-HTTP, also exist. Thus the maturity of the different individual technologies must be assessed individually.

2.3.3.2.1.1.1.4 (U) Standards

(U) Table 2.3-1 summarizes pertinent application and traditional application security standards discussed in this section.

Table 2.3-1: (U) Traditional Layered Application Security Standards

This table is (U)				
Reference	**Forum**	**Standards**	**Date**	**Maturity**
[SMTP]	IETF	RFC 821: Simple Mail Transfer Protocol	August 1982	Standard
[MSGFMT]	IETF	RFC 822: Standard for the Format of ARPA Internet Text Messages	August 1982	Standard
[PEM]	IETF	RFC 1421: Privacy Enhancement for Internet Electronic Mail: Part I: Message Encryption and Authentication Procedures	February 1993	Proposed Standard
	IETF	RFC 1422: Privacy Enhancement for Internet Electronic Mail: Part II: Certificate-Based Key Management	February 1993	Proposed Standard
	IETF	RFC 1423: Privacy Enhancement for Internet Electronic Mail: Part III: Algorithms, Modes, and Identifiers	February 1993	Proposed Standard
	IETF	RFC 1424: Privacy Enhancement for Internet Electronic Mail: Part IV: Key Certification and Related Services	February 1993	Proposed Standard
[MOSS]	IETF	RFC 1848: MIME Object Security Services	October 1995	Proposed Standard
[CMS]	IETF	RFC 3852: Cryptographic Message Syntax (CMS)	July 2004	Proposed Standard
[MSG]	IETF	RFC 3851: S/MIME v3.1 Message Specification	July 2004	Proposed Standard
[CERT]	IETF	RFC 3850: S/MIME v3.1 Certificate Handling	July 2004	Proposed Standard
[ESS]	IETF	RFC 2634: Enhanced Security Services for S/MIME	June 1999	Proposed Standard
	IETF	RFC 3854: Securing X.400 Content with S/MIME	July 2004	Proposed Standard
	IETF	RFC 3855: Transporting S/MIME Objects in X.400	July 2004	Proposed Standard
	IETF	RFC 3370: CMS Algorithms	August 2002	Proposed Standard
[CMC]	IETF	RFC 2797: Certificate Management Messages over CMS	April 2000	Proposed Standard
[HTTP]	IETF	RFC 2616: Hypertext Transfer Protocol -- HTTP/1.1	June 1999	Draft Standard
[HTTPAuth]	IETF	RFC 2617: HTTP Authentication: Basic and Digest Access Authentication	June 1999	Draft Standard
[S-HTTP]	IETF	RFC 2660: The Secure HyperText Transfer Protocol	August 1999	Experimental

This table is (U)				
Reference	**Forum**	**Standards**	**Date**	**Maturity**
[WebDAV}	IETF	RFC 2518: HTTP Extensions for Distributed Authoring -- WEBDAV	February 1999	Proposed Standard
[WebDAV-AC]	IETF	RFC 3744: WebDAV Access Control Protocol	May 2004	Proposed Standard
[SASL]	IETF	RFC 2222: Simple Authentication and Security Layer (SASL)	October 1997	Proposed Standard
	IETF	RFC 2444: The One-Time-Password SASL Mechanism	October 1998	Proposed Standard
	IETF	RFC 2554: SMTP Service Extension for Authentication	March 1999	Proposed Standard
[POP3]	IETF	RFC 1939: Post Office Protocol - Version 3	May 1996	Standard
	IETF	RFC 2449: POP3 Extension Mechanism	November 1998	Proposed Standard
	IETF	RFC 1734: POP3 AUTHentication command	December 1994	Proposed Standard
	IETF	RFC 3206: The SYS and AUTH POP Response Codes	February 2002	Proposed Standard
[IMAP4]	IETF	RFC 3501: Internet Message Access Protocol (IMAP) - Version 4rev1	March 2003	Proposed Standard
	IETF	RFC 2195: IMAP/POP AUTHorize Extension for Simple Challenge/Response	September 1997	Proposed Standard
	IETF	RFC 1731: IMAP4 Authentication Mechanisms	December 1994	Proposed Standard
	IETF	RFC 2086: IMAP4 ACL extension	January 1997	Proposed Standard
	IETF	RFC 2228: FTP Security Extensions	October 1997	Proposed Standard
[ACAP]	IETF	RFC 2244: Application Configuration Access Protocol	November 1997	Proposed Standard
[X.400]	ITU-T	X.400: Information Technology – Message Handling Systems (MHS) – Message Handling System and Service Overview	June 1999	Final Recomm.
[X.402]	ITU-T	X.402: Information Technology – Message Handling Systems (MHS) – Overall Architecture	June 1999	Final Recomm.
[X.411]	ITU-T	X.411: Information Technology – Message Handling Systems (MHS) – Message transfer system: Abstract Service Definition and Procedures	June 1999	Final Recomm.
[MSP]	NSA	SDN.701: Message Security Protocol	June 1996	v4.0
[CSP]	CCEB	ACP 120: Common Security Protocol (CSP)	June 1998	Base Edition

This table is (U)

Reference	Forum	Standards	Date	Maturity
[PKCS7]	RSA	PKCS #7: Cryptographic Message Syntax Standard	November 1993	v1.5

This table is (U)

2.3.3.2.1.1.1.5 (U) Dependencies

(U) Traditional application security technologies rely extensively on cryptographic technologies to provide encryption, digital signature, hash, and key exchange algorithms.

(U) Protocol development is a key enabling technology for traditional application security. At present, the dominant techniques rely on the following technologies:

- (U) Object-oriented design based on modeling in Abstract Syntax Notation One (ASN.1)
- (U) Syntactic description using Augmented Backus-Naur Format (ABNF)
- (U) Description of eXtensible Markup Language (XML) syntax using XML-Schema techniques
- (U) Formal system state analysis and modeling
- (U) Other formal techniques.

(U) These combined with a liberal application of English descriptive and ad-hoc techniques form a necessary part of application security development.

2.3.3.2.1.1.2 (U) Session Security Technologies

(U) As an alternative to developing application security, many applications choose to rely on session security technology. Session security technology protects all user data passed over a virtual connection between peer applications. Implementation of session security technology varies, and can be modeled variously as part of the application layer, or part of the transport layer. Session security technologies afford many of the same protections to user information, but with reduced flexibility and perhaps often with less permanence. Session security technologies are, however, vastly simpler than traditional layered application security and frequently offer rapid integration via exposed APIs.

2.3.3.2.1.1.2.1 (U) Technical Detail

2.3.3.2.1.1.2.1.1 (U) Secure Sockets Layer & Transport Layer Security

(U) The Secure Sockets Layer began as a proprietary technology developed by Netscape. SSL provided an extension to the popular Berkeley Sockets and Windows Sockets API to allow applications to invoke security services provided by a common encapsulation protocol. Initially, SSL was developed to service HTTP exclusively. Eventually it began to be used by broader range of applications.

(U) As SSL use became widespread, an effort was made to open the protocol and API definition to industry. This led to the development of the Transport Layer Security (TLS) standard in the IETF. TLS v1.0 is based on the SSL v3.0, but the two protocols do not interoperate. TLS implementations can, however, fall back to SSL 3.0 during negotiation. TLS v1.0 offers more flexibility in features and cryptography than SSL v3.0 and is expected to be the platform for all future evolution and development of the technology.

(U) TLS works by using the TLS Record Protocol to fragment data into manageable blocks. Each block has a MAC code applied, is (optionally) encrypted, and the resulting block is transmitted via TCP. Record Protocol might also compress the fragmented data—depending on the specific implementation. TLS uses the Record Protocol as a foundation for different types of protocol exchanges. The basic TLS specification defines four record types. Additional record types are supported as an extension mechanism. The following types are defined:

- (U) Handshake Protocol – Enables mutual establishment of identity between the client and server, and for negotiation of TLS options
- (U) Alert Protocol – Conveys information about important events in the communication such as normal closure of the association and errors
- (U) Change Cipher Spec Protocol – Enables the client and server to signal and mutually acknowledge transitions in ciphering strategies
- (U) Application Data Protocol – Conveys the fragmented, compressed, and encrypted application data. Messages are treated as transparent data.

(U) Although three of these protocols are quite simple, the TLS Handshake Protocol uses several staged exchanges. Figure 2.3-4 illustrates the context and operation of TLS and the Handshake Protocol.

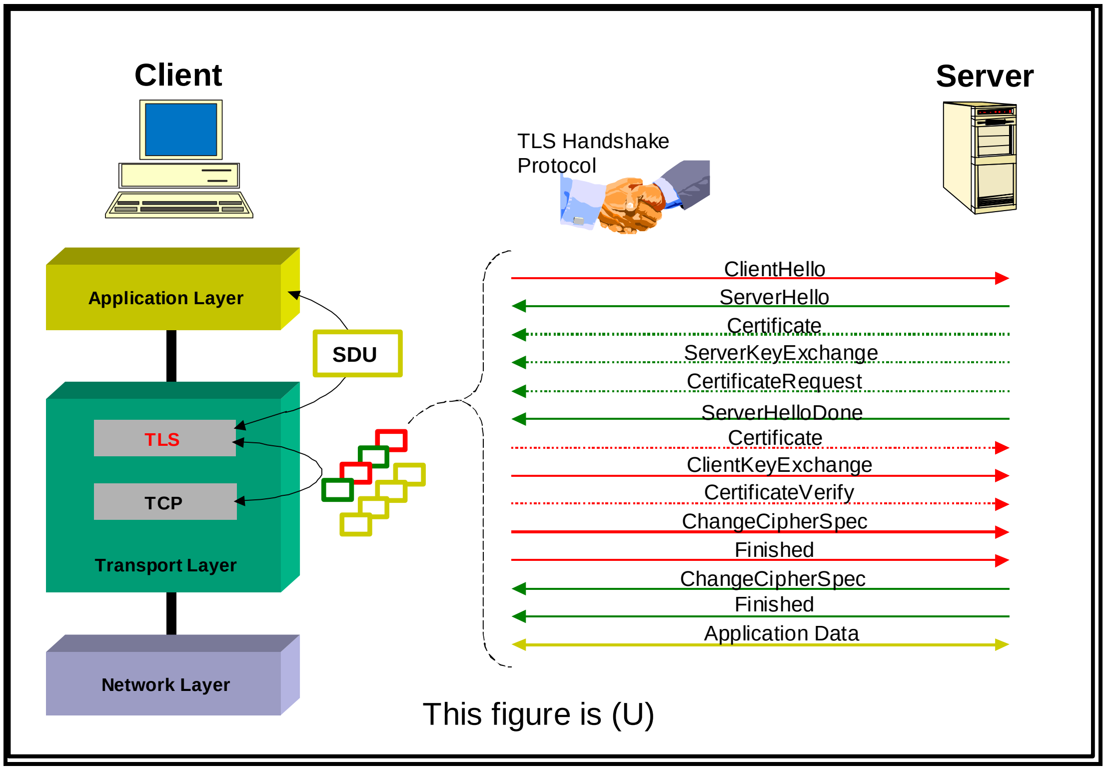


Figure 2.3-4: (U) TLS Handshake Protocol

(U) The TLS Handshake Protocol involves the following steps:

- (U) Exchange hello messages to agree on algorithms, exchange random values, and check for session resumption
- (U) Exchange the necessary cryptographic parameters to allow the client and server to agree on a pre-master secret
- (U) Exchange certificates and cryptographic information to allow the client and server to authenticate themselves
- (U) Generate a master secret from the pre-master secret and exchanged random values
- (U) Provide security parameters to the record layer
- (U) Allow the client and server to verify that their peer has calculated the same security parameters and that the handshake occurred without tampering.

(U) While the average user experience with TLS has mainly to do with confidentiality and integrity, the protocol is capable of strong mutual authentication. Authentication is only as strong as the Public Key Infrastructure (PKI) underlying the certificates issued to the client and server. While TLS-enabled servers commonly have certificates issued for their domains, most web browser implementations using TLS do not. Such browsers commonly establish an anonymous, but encrypted association with the TLS server and then perform basic authentication within that virtual circuit in accordance with HTTPAuth. When properly provisioned with certificates, TLS is capable of providing the following security services to an application:

- (U) Authentication of Server Identity
- (U) Authentication of Client Identity
- (U) Data Confidentiality
- (U) Data Integrity.

(U) TLS has been successfully applied to several different applications including:

- (U) HTTP (see HTTPTLS)
- (U) LDAPv3 (see LDAPAuth and LDAPTLS)
- (U) POP (see RFC 2595)
- (U) IMAP (see RFC 2595)
- (U) ACAP (see RFC 2595)
- (U) SMTP (see SMTPTLS).

2.3.3.2.1.1.2.1.2 (U) Generic Upper Layer Security

(U) In the early 1990s, the International Organization for Standardization (ISO) and International Telecommunication Union Telecommunication Standardization Sector (ITU-T) began to recognize a gap between the requirements for applications security set forth in CCITT X.800 | ISO/IEC 7498-2 (see OSISecArc]) and ITU-T X.803 | ISO/IEC 10745 (see ULSecMode]) and the practice of building security into individual applications from scratch. This realization led eventually to the development of the Generic Upper Layer Security (GULS) standards. GULS provided a set of standardized ASN.1 conventions to facilitate development of secure application syntaxes. It also defined a Security Exchange Service Element (SESE), which would establish and maintain a secure association over which application data could be exchanged securely. The SESE would function somewhat similarly to TLS. Unlike TLS, GULS was unambiguously modeled in the application layer and was distinct from OSI transport layer security standards.

(U) Unfortunately, GULS was of little value to existing OSI applications (e.g., X.400 and X.500) without modification. Also, since GULS was unambiguously wedded to ASN.1 and the OSI application layer structures, it was only of value to OSI applications. The total collapse of interest in OSI development in the mid-1990s virtually eliminated any work on new OSI applications or updates to existing applications. These factors have combined to make GULS virtually irrelevant today.

2.3.3.2.1.1.2.1.3 (U) Summary

(U) Session security technologies provide a very simple and potent solution for securing application communication. The development of TLS has proven extremely effective on widespread deployments and has been applied to a variety of applications. However, there are a number of limitations and security concerns on use of TLS for application security. GULS is of little present-day interest, but the similarity of evolution between GULS and TLS is noteworthy from the perspective of examining session security technologies as a whole.

2.3.3.2.1.1.2.2 (U) Usage Considerations

(U) Caution should be exercised in employing session security technologies, such as TLS, for application security purposes. The suitability of TLS depends heavily on it functioning in an overall security architecture. For example, TLS can be subjected to man-in-the-middle attacks. So care must be taken that strong 2-way authentication is applied during the Handshake Protocol, and that certificates or other credentials are validated and recognized. This is true even if subsequent access control based on [HTTPAuth] with be used within the TLS association. TLS is also vulnerable to compromise of its feature negotiation mechanisms. So care must be taken to ensure that the implementation minimum acceptable security measures reflect the security policy in force. TLS is also not suitable for application architectures that require secure multipoint communications, multiple different application entities or architectures that require persistent security that endures through a relaying application entity.

2.3.3.2.1.1.2.3 (U) Maturity

(U) Session security technologies are Mature (TRLs 7 – 9), and TLS in particular is a Mature, widely implemented, and well deployed solution. It is worth noting that most TLS client implementations operate without certificates or public keys by default. Most are not easily configurable to employ a per-application certificate much less a per-user certificate. Therefore it seems likely that more product improvement must take place for TLS to expand beyond web browsing and properly provide security to multiple applications.

2.3.3.2.1.1.2.4 (U) Standards

(U) Table 2.3-2 summarizes pertinent session security standards discussed in this section.

Table 2.3-2: (U) Session Security Standards

This table is (U)

Reference	Forum	Standards	Date	Maturity
[TLS]	IETF	RFC 2246: The TLS Protocol v1.0	January 1999	Proposed Standard
[HTTPTLS]	IETF	RFC 2817: Upgrading to TLS Within HTTP/1.1	May 2000	Proposed Standard
	IETF	RFC 2818: HTTP Over TLS	May 2000	Informational
[TLSEXT]	IETF	RFC 3546: TLS Extensions	June 2003	Proposed Standard
[AESTLS]	IETF	RFC 3268: AES Ciphersuites for TLS	June 2002	Proposed Standard
[LDAPAuth]	IETF	RFC 2829: Authentication Methods for LDAP	May 2000	Proposed Standard
[LDAPTLS]	IETF	RFC 2830: LDAPv3 Extension for TLS	May 2000	Proposed Standard
[LDAPv3]	IETF	RFC 3377: LDAP v3 Technical Specification	September 2002	Proposed Standard
[RFC2595]	IETF	RFC 2595: Using TLS with IMAP, POP3 and ACAP	June 1999	Proposed Standard
[SMTPTLS]	IETF	RFC 3207: SMTP Service Extension for Secure SMTP over TLS	February 2002	Proposed Standard
[GULS]	ISO	ISO/IEC 11586-1: Information technology -- Open Systems Interconnection -- Generic upper layers security: Overview, models and notation	1996	International Standard
	ISO	ISO/IEC 11586-2: Information technology -- Open Systems Interconnection -- Generic upper layers security: Security Exchange Service Element (SESE) service definition	1996	International Standard
	ISO	ISO/IEC 11586-3: Information technology -- Open Systems Interconnection -- Generic upper layers security: Security Exchange Service Element (SESE) protocol specification	1996	International Standard
	ISO	ISO/IEC 11586-4: Information technology -- Open Systems Interconnection -- Generic upper layers security: Protecting transfer syntax specification	1996	International Standard

This table is (U)				
Reference	**Forum**	**Standards**	**Date**	**Maturity**
	ISO	ISO/IEC 11586-5: Information technology -- Open Systems Interconnection -- Generic upper layers security: Security Exchange Service Element (SESE) Protocol Implementation Conformance Statement (PICS) proforma	1997	International Standard
	ISO	ISO/IEC 11586-6: Information technology -- Open Systems Interconnection -- Generic upper layers security: Protecting transfer syntax Protocol Implementation Conformance Statement (PICS) proforma	1997	International Standard
[OSISecArch]	ISO	ISO/IEC 7498-2: Data Communication Networks – Open Systems Interconnection (OSI) – Security, Structure and Applications – Security Architecture for Open Systems Interconnection for CCITT Applications	1989	International Standard
[ULSecModel]	ISO	ISO/IEC 10745: Information Technology – Open Systems Interconnection – Upper Layers Security Model	July 1994	International Standard
[OSISecArch]	ITU-T	CCITT X.800: Data Communication Networks – Open Systems Interconnection (OSI) – Security, Structure and Applications – Security Architecture for Open Systems Interconnection for CCITT Applications	1991	Final Recomm.
[ULSecModel]	ITU-T	ITU-T X.803: Information Technology – Open Systems Interconnection – Upper Layers Security Model	July 1994	Final Recomm.
[GULS]	ITU-T	ITU-T X.830: Information technology -- Open Systems Interconnection -- Generic upper layers security: Overview, models and notation	April 1995	Final Recomm.
	ITU-T	ITU-T X.831: Information technology -- Open Systems Interconnection -- Generic upper layers security: Security Exchange Service Element (SESE) service definition	April 1995	Final Recomm.
	ITU-T	ITU-T X.832: Information technology -- Open Systems Interconnection -- Generic upper layers security: Security Exchange Service Element (SESE) protocol specification	April 1995	Final Recomm.

This table is (U)

Reference	Forum	Standards	Date	Maturity
	ITU-T	ITU-T X.833: Information technology -- Open Systems Interconnection -- Generic upper layers security: Protecting transfer syntax specification	April 1995	Final Recomm.
	ITU-T	ITU-T X.834: Information technology -- Open Systems Interconnection -- Generic upper layers security: Security Exchange Service Element (SESE) Protocol Implementation Conformance Statement (PICS) proforma	October 1996	Final Recomm.
	ITU-T	ITU-T X.835: Information technology -- Open Systems Interconnection -- Generic upper layers security: Protecting transfer syntax Protocol Implementation Conformance Statement (PICS) proforma	October 1996	Final Recomm.

This table is (U)

2.3.3.2.1.1.2.5 (U) Dependencies

(U) Neither cryptography nor security protocol development are discussed in detail in this section. However, session security technologies have a similar dependency on them.

2.3.3.2.1.1.3 (U) Web Services Security Technologies

(U) The future of application development for the GIG is expected to take a different direction from past application layer development. The emphasis for GIG applications is expected to be service-oriented architectures. And the primary focus for service-oriented application development is the technology known as Web Services. Unfortunately, development security technology for Web Services is still in its infancy.

2.3.3.2.1.1.3.1 (U) Technical Detail

(U) With the tremendous success of web browsing as the Internet's second killer application, pressure grew to leverage the success and ubiquity of the web for other purposes. Recognition also dawned that while HTTP servers and dynamically generated HTML documents were sufficient to allow humans users basic access to databases, they were not sufficient to enable automated systems to access information in those same databases. This was a function of HTML being optimized for specifying presentation rather than semantics. This led to the development of the XML, which was optimized instead for identifying the semantics of data.

(U) In the late 1990s, the Simple Object Access Protocol (SOAP) was developed as a means to allow XML objects to be requested and transferred over HTTP or a variety of other protocols. SOAP provides an XML envelope consisting of a heading and body. The specification in SOAP provides bindings between SOAP and HTTP so that SOAP transactions can take advantage of the existing, ubiquitous HTTP infrastructure. Other bindings, such as to SMTP or other existing protocols, are also possible but seldom seen. Services built to request and delivery specific data using XML and SOAP have come to be known as Web Services.

(U) Developing security services as a common add-on to the web services framework offer significant benefits over traditional layered application security development. Figure 2.3-5 contrasts the web services model with that shown previously for CMS and S/MIME. In the web services framework a variety of service offerings can be provided through SOAP and HTTP. Each service would benefit from the same security elements applied to the common SOAP envelope. This form of security is called Web Services Security (WSS). Conceptually, WSS has much in common with a reusable module, such as CMS, or session security services, such as TLS. However, WSS has the potential to combine the best elements of both.

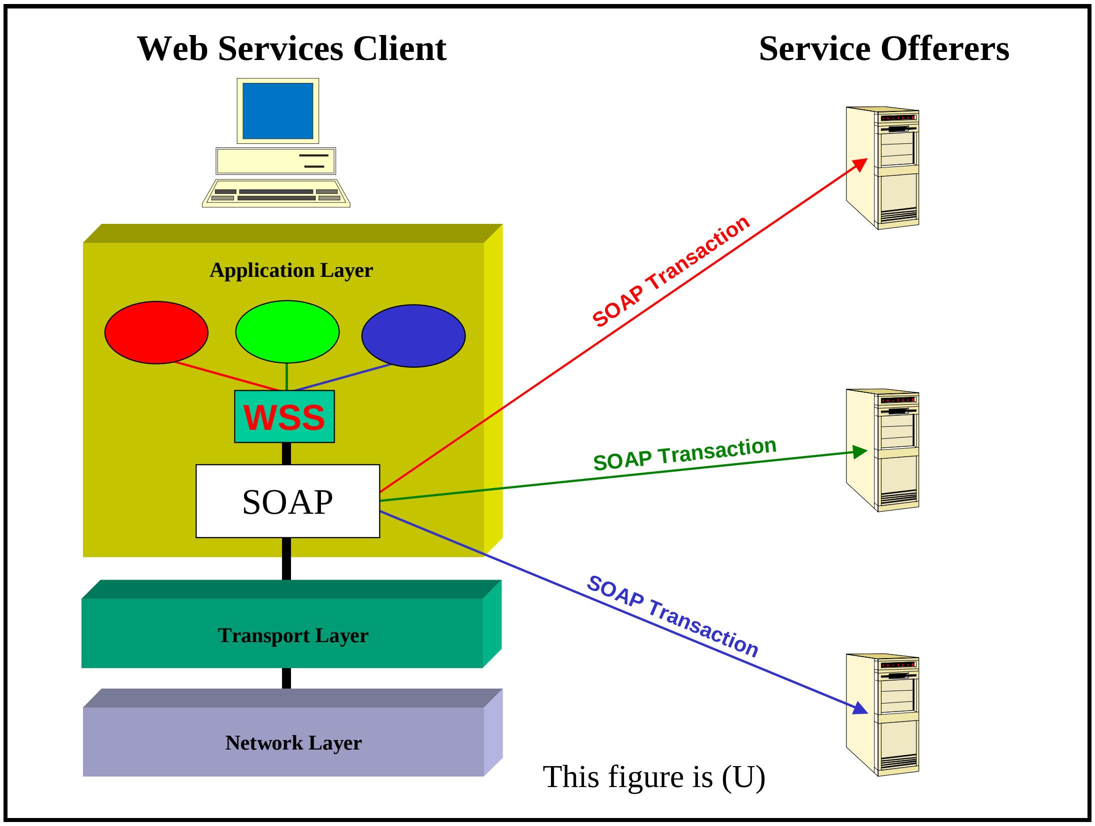


Figure 2.3-5: (U) Model for Web Services Security

(U) Different organizations are involved in developing standards and specifications for WSS. The World Wide Web Consortium (W3C), the organization responsible for the original development of both XML and SOAP, has contributed to the development of WSS by introducing the XML Digital Signature (XML_DSIG) and XML Encryption (XML_ENC) standards. These have the potential to become foundation standards for more advanced WSS development. In competition with the W3C standards is the work of the American National Standards Institute (ANSI). ANSI has developed an XML Cryptographic Message Syntax (XCMS) which provides functions similar to XML_DSIG and XML_ENC, but does so by applying a relatively simple XML wrapper to the existing IETF CMS wrappers. It is unclear at this point which approach will dominate.

(U) The Organization for the Advancement of Structured Information Standards (OASIS) is developing several standards that have the promise to contribute to WSS. These include SAML, XACML, and WSS.

(U) Another significant WSS development is under way at the Web Services Interoperability (WS-I) Organization. WS-I is engaged in an effort to achieve commonality and interoperability among web service components. WS-I has already released the WS-I Basic Profile for web services and is continuing work on a draft Basic Security Profile for WSS.

(U) Another contender in the WSS area is Liberty Alliance. They are focused on solving the problem of cooperation between federated web services to provide secure operation where all of the participants may not be part of the same organization or necessarily share a common security policy. It is unclear how the Liberty Alliance work will ultimately affect the overall WSS effort. Liberty Alliance has released three sets of standards that promise to have an impact on WSS.

- (U) The Identity Federation Framework (ID-FF) offers an approach for establishing a standardized, multi-vendor, web-based SSO with federated identities based on commonly deployed technologies
- (U) The Identity Web Services Framework (ID-WSF) is a set of specifications for creating, using, and updating various aspects of identities
- (U) The Identity Services Interface Specifications (ID-SIS) define profiles for commonly useful services, including a personal profile service (ID-SIS-PP) that provides basic profile information such as contact information and an employee profile service (ID-SIS-EP) that provides Employee's basic profile information.

2.3.3.2.1.1.3.2 (U) Maturity

(U) WSS standards are Emerging (TRLs 4 – 6). They are still under development and are not ready for full scale deployment. Further, there are different standards competing for many of the same functional requirements. It is not clear at this point which standards will succeed and in what market segments. It is possible that some security standards will prove to be suited to certain types of web service while others will better support different forms of web service. So there is considerable risk in early adoption of any of these immature solutions.

2.3.3.2.1.1.3.3 (U) Standards

(U) Table 2.3-3 summarizes pertinent web services security standards discussed in this section.

Table 2.3-3: (U) Web Services Security Standards

This table is (U)

Reference	Forum	Standards	Date	Maturity
[XML]	W3C	XML		Final
		XML Schema		Stable
[XML-DSIG]		XML-DSIG		Final
[XML-ENC]		XML-ENC		Final
		XKMS		Revision
[SOAP]		SOAP		Revision
		WSDL		Revision
[SAML]	OASIS	SAML		Stable
[XACML]		XACML		Revision
		UDDI		Revision
		SPML		Stable
		XCBF		Final

This table is (U)

Reference	Forum	Standards	Date	Maturity
		XCBF Token Profile		Final
[WSS]		Web Services Security (WSS)		Revision
		WSS UsernameToken Profile		Revision
		WSS X.509 Certificate Token Profile		Revision
		Web Services Reliable Messaging		Draft
		ebXML Registry		
		ebSOA		
		WSDM		
		XrML (eXtensible Rights Management Language)		Draft
		Web Application Security		
		Digital Signature Services		
		Security Services		
		Web Services Distributed Management		
[WSI-SEC]	WS-I	Basic Security Profile Security Scenarios		Draft
		Basic Profile		Revision
	ANSI	ANSI X9.84 (XCBF)		Final
[XCMS]		ANSI X9.96 (XCMS)		
		ANSI X9.73 (CMS)		
	ITU-T	ITU-T X.509		
	ISO	ISO 19092 (biometric formats)		Draft
[ID-FF]	Liberty Alliance	ID-FF		Stable
[ID-SIS]		ID-SIS		Revision
[ID-WSF]		ID-WSF		Revision
		draft-lib-arch-soap-authn		Draft

This table is (U)

2.3.3.2.1.1.3.4 (U) Dependencies

(U) Neither cryptography nor security protocol development are discussed in detail in this section. However, web services security technologies have a similar dependency on them. It should be noted that web services' exclusive focus on SOAP and XML narrow the range of techniques used in security protocol development.

2.3.3.2.1.2 (U) Real-Time Data Technologies

2.3.3.2.1.2.1 (U) FNBDT

2.3.3.2.1.2.1.1 (U) Technical Detail

(U) Future Narrowband Digital Terminal (FNBDT) is a group of signaling and cryptography specifications designed to allow end-to-end secure communications using commercial communications channels. FNBDT operates at the Application Layer (see Figure 2.3-6) and is designed to operate over whatever transport method is available.

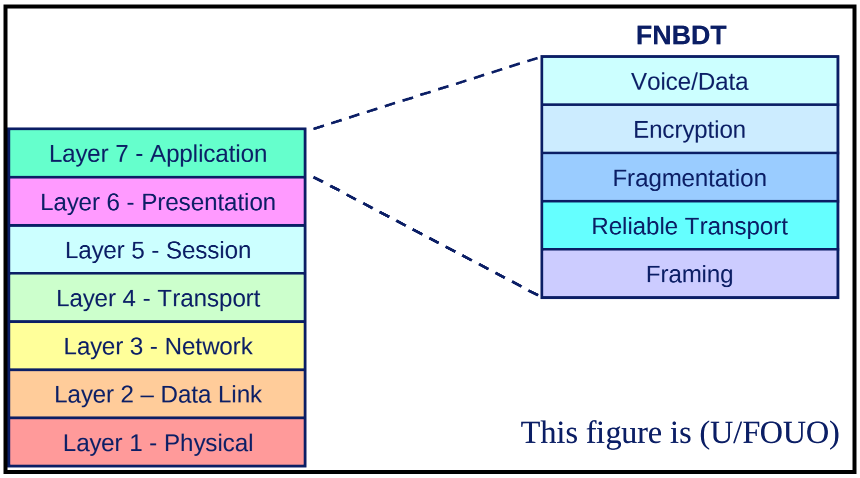


Figure 2.3-6: (U) FNBDT Location in Network Protocol Stack

(U//FOUO) FNBDT specifications define the following aspects of secure voice and data communication:

- (U//FOUO) The signaling required to establish and maintain secure calls independent of the transport network
- (U//FOUO) A Minimum Essential Requirement (MER) mode which guarantees interoperability between FNBDT-compliant devices
- (U//FOUO) Key management for generating and maintaining compatible encryption keys
- (U) Encryption algorithms
- (U//FOUO) MELP (2400 bps) and G.729D (6400 bps) voice coders
- (U//FOUO) Cryptographic synchronization management functionality
- (U//FOUO) An escape mechanism enabling venders to implement proprietary modes.

(U//FOUO) Currently the FNBDT specifications specify only Type 1 encryption methods, although the signaling is directly applicable to vendor-defined non-Type 1 applications. Multiple vendors have introduced Type 1 and non-Type 1 products based on the FNBDT specifications.

(U//FOUO) FNBDT provides the ability for products to operate in high Bit Error Rate (BER) environments. Establishing an FNBDT channel involves an initial negotiation of capabilities between endpoints, with the ability to select vendor proprietary modes if both endpoints have compatible capabilities. Compatible operational modes, encryption algorithms, and key sets are also selected during this initial exchange.

2.3.3.2.1.2.1.2 (U) Usage Considerations

2.3.3.2.1.2.1.2.1 (U) Implementation Issues

(U//FOUO) The FNBDT signaling protocol at the Application Layer has proved to be a successful method of providing security for voice systems. While the FNBDT protocol is not oriented toward a packet-based system, it does not inherently prohibit operating with such a system. FNBDT is a streaming protocol that defines a constant-rate bitstream. For voice applications, this bitstream is either 2400 bps or 7200 bps. As long as the receiving end of the communication link can receive the bits and reformat them into the same constant-rate bitstream that was presented to the network at the transmit end of the link, the FNBDT signaling protocol will be adequate for secure voice applications.

(U) Packet-based transport systems present unique challenges for streaming protocols such as FNBDT. The following list identifies several sources of degradation introduced by packet-based systems and evaluates the tolerance of the FNBDT signaling protocol to these degradation sources.

(U//FOUO) Packet latency. This refers to the network delay in transporting bits from one end to the other. Two-way real-time applications such as voice conversations are negatively affected by total delay times that are perceptible to the user, typically in the 0.5 sec range. Because there are other sources of delay in the system besides packet transport time, the delay introduced by packet transport must be significantly less than this. The FNBDT protocol is not inherently affected by increased packet latency, although of course the regenerated speech at the receiving end of the link will be delayed accordingly.

(U//FOUO) Packet jitter. Packet jitter refers to the difference in time required to transport packets, as opposed to the absolute delay (packet latency). Streaming protocols such as FNBDT are required to maintain a constant-rate output even when the network transport mechanism results in packets arriving at different times. This is typically resolved by buffering at the receive end of the link. Packets are fed into the buffer at varying times as they arrive, but are read out of the buffer at the constant (streaming) rate required by the application, a process shown in Figure 2.3-7. The buffer must be able to accommodate the largest potential jitter, and therefore the net result of this arrangement is that the received signal is delayed enough to account for the largest potential jitter. This delay is in addition to the delay introduced by packet latency. As with packet latency, the FNBDT protocol is not inherently affected by increased jitter.

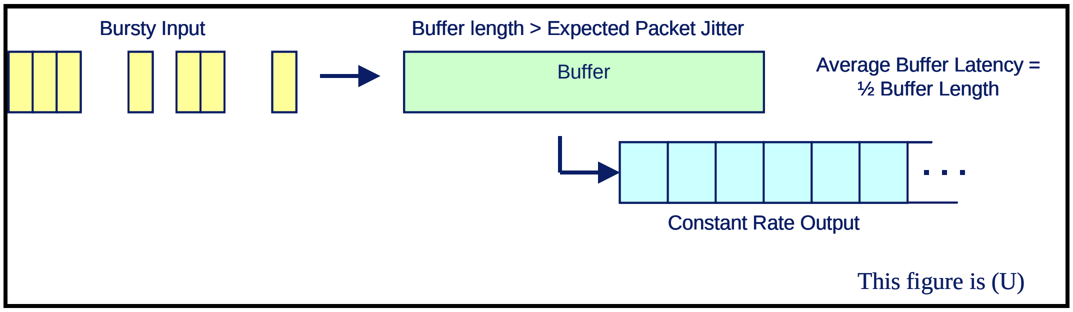


Figure 2.3-7: (U) Packet Jitter Mitigation Process

(U) Packet loss. Packets may be lost during the transport process, resulting in missing data at the receiver. In the case of secure applications, missing data invariably leads to loss of cryptographic synchronization. Any subsequent data received and decrypted will be garbled until cryptographic synchronization is re-established. This potentially devastating situation is mitigated by the FNBDT signaling protocol, which includes embedded cryptographic synchronization information periodically in the transmitted bitstream. This cryptographic re-synchronization information occurs every 320 msec for G.729D speech and every 540 msec for MELP speech. The potential impact of individual lost packets is therefore a short (0-500 msec) section of garbled speech during a conversation. Periods of sequential lost packets will result in appropriately long periods of missing or garbled speech, with a 0-500 msec period for re-establishing cryptographic synchronization when the packets begin arriving again.

(U) Packet re-ordering. Some packet transport systems have the capability to use different paths for transporting packets, resulting in the potential for packets to arrive out of order. Often the transport system has the capability to rearrange the received packets to the correct order before presenting them to the upper layers, resulting in packet jitter rather than actual ordering errors in the bitstream. If, however, information is presented to an FNBDT receiver with segments out of order, the out-of-order segments will result in random (garbled) information. The length of any such garbled data will depend on the packet size. Since speech applications will likely keep packets small in order to reduce latency, the period of speech degradation will likewise be small. Packet re-ordering issues lead to cryptosync loss with appropriate recovery periods as described in the previous paragraph.

(U//FOUO) Packet bit-errors. Uncorrected bit errors within transmitted packets will have the same effect as bit errors in a circuit switched network. The FNBDT protocol was designed for relatively high bit-error rate environments (~2%) and includes automatic retransmission capabilities for those portions of the signaling which must arrive error-free. Once a secure call is established, the speech algorithms themselves are extremely tolerant to random bit errors. Individual bit errors seldom result in noticeable degradation to the received speech. FNBDT traffic modes use crypto methods that do not result in bit error extension, meaning that single bit errors in the received ciphertext do not extend to multiple bit errors in the decrypted plaintext.

2.3.3.2.1.2.1.2.2 (U) Advantages

(U//FOUO) FNBDT is an end-user to end-user protocol. Information is encrypted at the transmitting end-user where traffic is generated and is never decrypted until it arrives at the receiving end-user where the traffic is consumed. User data is protected through whatever network and across whatever communication channels might be traversed. Where gateways are required to deliver bits from one protocol stack to another (e.g., VoIP to PSTN) user data remains encrypted as it traverses the gateway.

(U//FOUO) The FNBDT protocol provides inherent transport reliability (Ack/Nak with retransmission) for signaling messages. Voice modes operate without any underlying retransmission protocol to reduce latency. Data modes are defined with and without retransmission to allow increased throughput (Guaranteed Throughput mode) or increased reliability (Reliable Transport mode) as required for specific applications. The frame structure for signaling transport reliability and Reliable Transport data mode is shown in Figure 2.3-8.

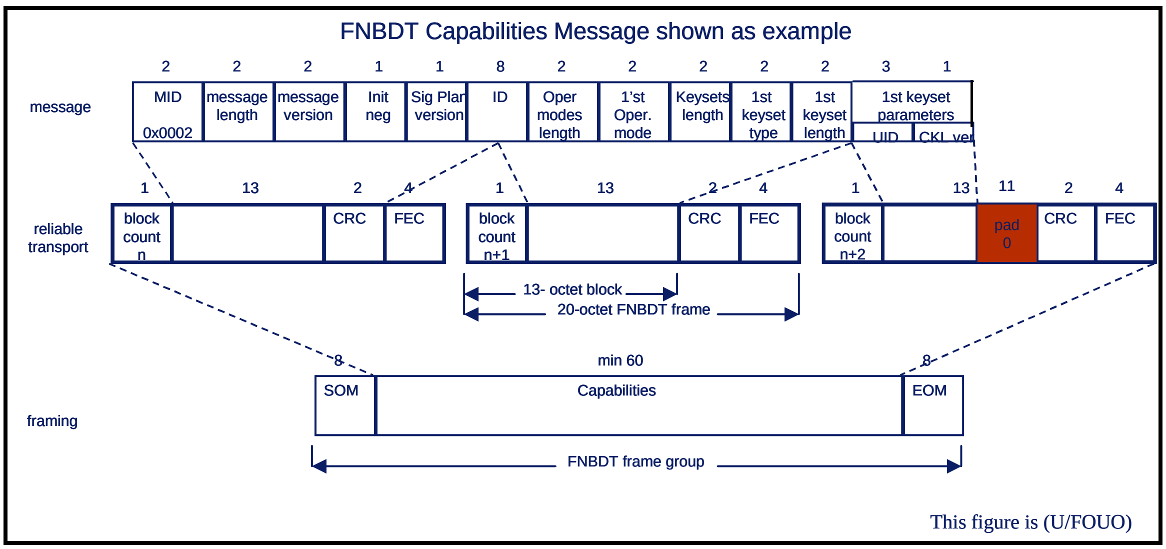


Figure 2.3-8: (U//FOUO) FNBDT Frame Structure for Signaling Reliability and Reliable Transport Data Mode

(U//FOUO) An inherent strength of the FNBDT protocol is its ability to maintain cryptographic synchronization for secure voice applications throughout signal fading and high BER environments. Without this ability, the application data would continually need to be interrupted to resynchronize the cryptography as data is lost or corrupted, leading to annoying gaps and artifacts in encrypted speech.

(U//FOUO) Synchronization is accomplished by periodically embedding information in the transmitted bitstream. This allows the receiver to resynchronize the cryptography without using channel resources other than the periodic embedded information. When the MELP vocoder has been selected, the FNBDT specifications define both a Blank and Burst mode where cryptographic resynchronization information replaces every 24th vocoder frame as indicated in Figure 2.3-9, and a Blank without Burst mode where all vocoder frames are transmitted. The Blank and Burst mode results in no additional overhead for the embedded resync information, which occur every 540 msec and results in a composite bitstream of 2400 bps.

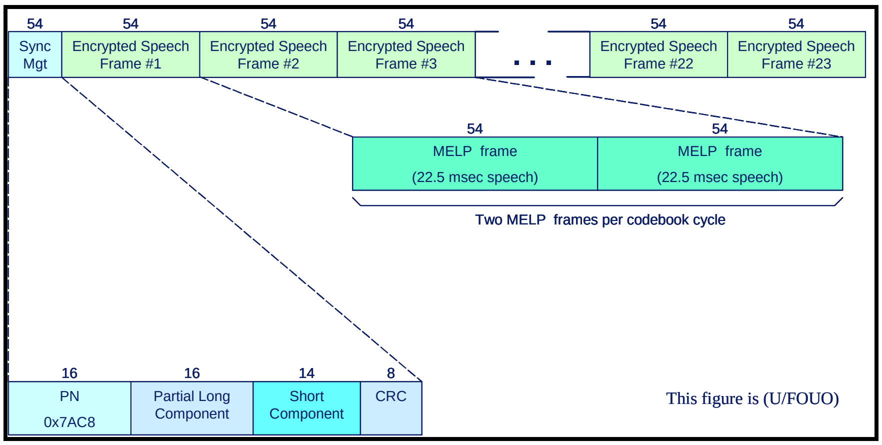


Figure 2.3-9: (U//FOUO) FNBDT 2400 bps MELP Blank and Burst Superframe Structure

(U//FOUO) When the G.729D vocoder has been selected, cryptographic resynchronization information is inserted every 8th vocoder frame as shown by Figure 2.3-10. This allows the cryptography to resynchronize every 320 msec and results in a composite bitstream of 7200 bps.

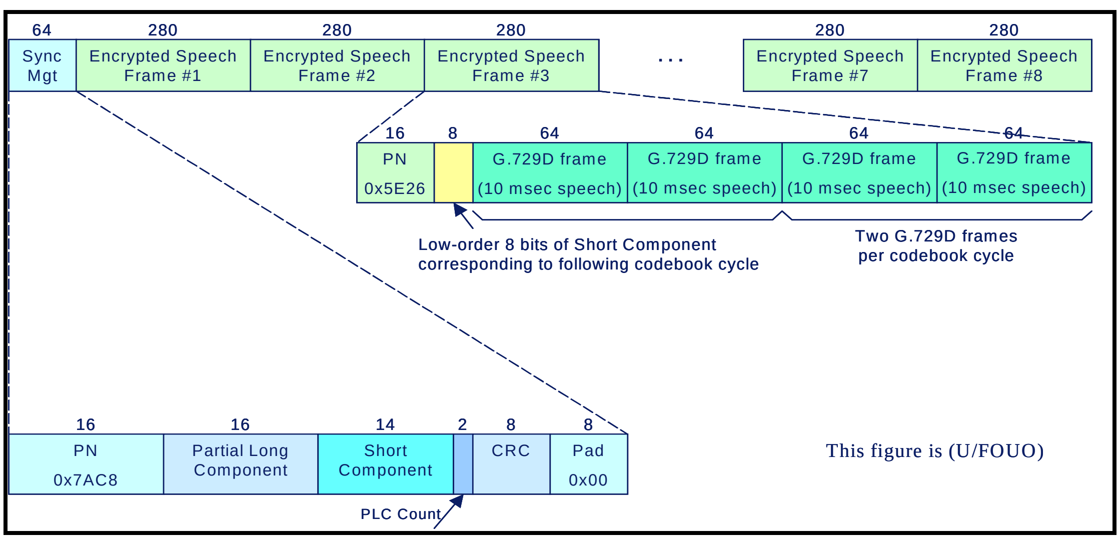


Figure 2.3-10: (U//FOUO) FNBDT 7200 bps G.729D Superframe Structure

(U//FOUO) FNBDT is particularly useful in high BER environments where channels are likely to fade and where low latency, real-time encrypted speech and data applications are required.

(U//FOUO) The FNBDT protocol is transport-independent in that it is designed to operate over whatever lower-layer protocols might be available. Within constraints applicable to specific applications (timeouts, speech delay, etc.) the FNBDT protocol can operate over any channel capable of transporting bits between two end-user terminals. FNBDT terminal vendors have implemented products utilizing PSTN, ISDN, GSM, CDMA, Iridium satellite, digital radio, and other channel types for data transport.

2.3.3.2.1.2.1.2.3 (U//FOUO) Risks/Threats/Attacks

(U//FOUO) Since the FNBDT protocol operates at the Application Layer in the network protocol stack, risks associated with lower protocol layers are not addressed. Issues such as Traffic Flow analysis, LPI/LPD, and DoS must be dealt with outside the bounds of the FNBDT protocols.

2.3.3.2.1.2.1.3 (U//FOUO) Maturity

(U//FOUO) The FNBDT protocol is Mature, in the sense that products have been implemented and deployed for several years. Users have real-world experience with FNBDT products—both wired and wireless. Additional modes and features will continue to be added to the specifications, but the basic interoperable FNBDT modes are mature and will continue to exist for some time into the future. The TRL of the basic FNBDT bitstream definition is 9 (Mature - products deployed in operational mission conditions).

(U//FOUO) Application of the FNBDT protocol to IP-based transport is less mature. Although different vendors are working to apply FNBDT technology to IP networks, there are currently no interoperable standards for this specific application. The TRL for using FNBDT over IP networks is currently estimated at 4 (Emerging - breadboard validation in lab environment).

2.3.3.2.1.2.1.4 (U) Standards

(U//FOUO) The FNBDT protocols are defined by the standards listed in Table 2.3-4:

Table 2.3-4: (U) FNBDT Standards

This table is (U//FOUO)

Name	Description
FNBDT-210 (Signaling Plan)	This unclassified specification defines the signaling requirements for FNBDT operational modes. A secure overlay capable of interoperation with FNBDT compatible equipment on various similar or disparate networks is defined. Since the various networks will often have different lower-layer communications protocols, the FNBDT secure overlay specification specifies the higher-layer end-to-end protocols only. Appendices to this specification define operation using specific networks.
FNBDT-230 (Cryptography Specification)	This classified specification outlines details of the cryptography defined for FNBDT. Issues such as key generation, traffic encryption, and compromise recovery are specified in sufficient detail to allow interoperable implementation.
Proprietary extensions	The FNBDT signaling and cryptography specifications define interoperable branch points allowing vendors to implement proprietary modes. This allows vendors to take advantage of the basic FNBDT structure to add modes fulfilling specific needs. Legacy FNBDT implementations have used these branch points to implement custom cryptographic modes. Details of such modes are contained in vendor proprietary specifications.

This table is (U//FOUO)	
Name	**Description**
Other specifications	Other interoperable FNBDT specifications have been suggested and are currently under consideration by the FNBDT Working Group. These additional documents would provide interoperable ways of implementing additional features such as non-Type 1 operation and key management.
This table is (U//FOUO)	

2.3.3.2.1.2.1.5 (U) Cost/Limitations

(U//FOUO) Although the FNBDT protocol is a good choice for solving many speech-related security issues, there are limitations with this protocol as well. Potential limiting factors that must be considered when evaluating FNBDT as a candidate protocol for solving security problems include:

- (U//FOUO) Point-to-point operation. The current definition of FNBDT includes point-to-point operation only. There are no provisions in place for multi-user conferencing or net broadcast capabilities. The FNBDT Working Group is currently active in defining net broadcast modes and Pre-Placed Key (PPK) methods allowing multiple users to decrypt a common encrypted bitstream.

- (U//FOUO) Voice coders. The FNBDT specifications currently define two voice coders; 2400 bps MELP and 6400 bps G.729D. FNBDT-compatible speech equipment must include one of these vocoders in order to interoperate with FNBDT equipment provided by other vendors.

- (U//FOUO) Legacy interoperability. FNBDT equipment is not compatible with other types of secure voice equipment. Specifically, the older generation STU-III devices that have been widely deployed throughout the world during the past 20 years are not compatible with the cryptography, speech coders, or wireline modems used by FNBDT equipment.

- (U//FOUO) Establishing a channel. FNBDT is defined as an application layer technology that provides the encrypted bitstream to transfer between two endpoints. The details regarding how the digital channel is established between these two endpoints is left outside the scope of the FNBDT specifications. As a result, potential users must be aware of channel establishment procedures to make sure this process is successful outside the bounds of FNBDT.

- (U//FOUO) Trusted platform requirement. Application Layer security methods are not suitable for operation using general purpose computing equipment. FNBDT and other Application Layer security approaches require trusted hardware to support separation requirements.

2.3.3.2.1.2.1.6 (U) Dependencies

(U//FOUO) FNBDT cryptography specifications depend on terminals containing appropriate key material. The necessary key material is supplied by the Government's Electronic Key Management System (EKMS).

(U//FOUO) The call control process (call establishment, maintenance, teardown, etc.) is not defined by the FNBDT protocol. These processes, which are a necessary part of a successful FNBDT voice or data call, must occur outside the scope of the FNBDT specifications.

2.3.3.2.1.2.1.7 (U) Alternatives

(U//FOUO) The most widespread alternative to FNBDT secure speech systems continues to be the STU-III terminals. These devices, which are based on approximately 20-year old technology, are no longer produced but are so pervasive throughout the Government that they continue to be a factor in secure speech system decisions. The Government expects to continue producing key material to support these terminals through the GIG 2008 Vision timeframe.

(U//FOUO) Other tactical and strategic secure voice system terminals exist in lower quantities. Systems such as Advanced Narrowband Digital Voice Terminal (ANDVT), MSE, etc. are also relatively dated but continue to provide acceptable quality encrypted speech communications for certain specific applications.

(U//FOUO) Depending on specific operational requirements, a speech channel could be protected at the IP layer (e.g., HAIPE) rather than the Application Layer. This approach, referred to as Voice over Secure IP rather than Secure Voice over IP, provides an alternative to the FNBDT Application Layer protection approach for user environments where separation within an enclave is not a consideration.

2.3.3.2.1.2.1.8 (U) Complementary Techniques

(U//FOUO) Any given user situation may require a combination of technologies in order to meet all operational requirements. For example, the FNBDT protocol may provide confidentiality at the Application Layer, but does nothing toward meeting any potential Traffic Flow Security or TRANSEC requirements at the lower layers. Additional technologies will often need to be used in combination with the FNBDT protocol in order to meet all applicable security requirements.

2.3.3.2.1.2.2 (U) Interoperability/Gateways

2.3.3.2.1.2.2.1 (U) Technical Detail

(U//FOUO) Interoperability is an important GIG consideration, both from enclave to enclave within the GIG and from GIG resources to infrastructure external to the GIG. Gateways provide the necessary interworking and protocol stack adaptation to provide this interoperability.

(U) Gateways adapt the communication needs of different networks such that user data can be sent from one to another. Gateways can be described as relay devices; that is, they relay user traffic from one protocol stack to another.

(U) Gateways can be grouped according to the specific functions they perform. Some are signaling gateways that adapt the call control and other signaling needs of a particular network to the signaling needs of a different network. Some are media gateways that adapt user speech from one form to another. Signaling and media functions can be combined such that a common device provides both functions.

(U//FOUO) Within the GIG architecture, gateways will be necessary both for providing interoperability between different vendors VoIP implementations and for providing interoperability between packet-switched and circuit-switched networks.

(U) Figure 2.3-11 illustrates the protocol stacks associated with a typical Media Gateway (MG) included with a VoIP system for interoperation with legacy PSTN networks. This MG provides a termination point for the IP, UDP, and RTP layers, as well as providing a transcoder function. The result is audio speech that can be routed to the PSTN.

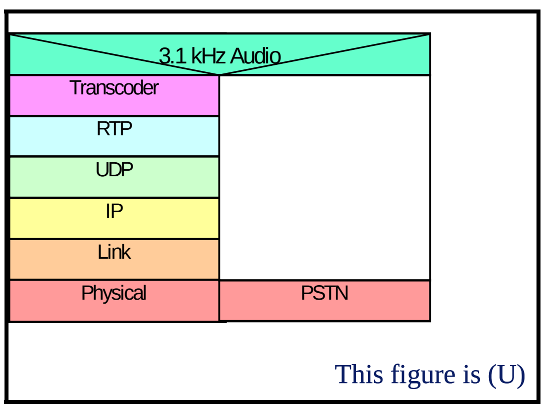


Figure 2.3-11: (U) Media Gateway Protocol Stack Illustration

(U//FOUO) Tactical networks within the GIG may also include gateway functionality to allow interoperation with other systems. Like the commercial VoIP gateways, these tactical versions contain a protocol stack appropriate to the specific tactical network on one side and a protocol stack appropriate to the target network on the other.

(U//FOUO) Although gateways will remain a necessary part of the infrastructure, it is important that secure system architectures are designed so that gateways remain Black. This means that although the gateways may remove or adapt network protocol stack layers, they must not be expected to decrypt user traffic. User traffic must remain encrypted as it traverses the gateway—resulting in true end-user to end-user encryption.

2.3.3.2.1.2.2.2 (U) Usage Considerations

2.3.3.2.1.2.2.2.1 (U) Implementation Issues

(U) Legacy PSTN-based secure voice systems transport bits using a commercial wireline modem to modulate digital traffic over the analog PSTN. In order for secure VoIP terminals to interoperate with these legacy systems, the gateway must provide a compatible modem function on the PSTN side. Although commercial VoIP systems today have recognized the need for PSTN Interworking and have included the Media Gateway functionality, there is no commonly recognized need to include the modem function in this gateway.

(U//FOUO) Therefore, non-standard gateways are required to allow interworking between secure VoIP systems and legacy secure PSTN-based systems. Although gateways containing this functionality have not been identified as a requirement in commercial VoIP systems, it is important to point out that implementation and maintenance of such a gateway does not necessarily need to be carried out by the same vendor that supplies the basic VoIP system. A system integrator having access to the IP network on one side and the PSTN on the other could insert the required gateway independent of the other infrastructure.

(U//FOUO) The functionality associated with a secure VoIP gateway is shown in Figure 2.3-12.

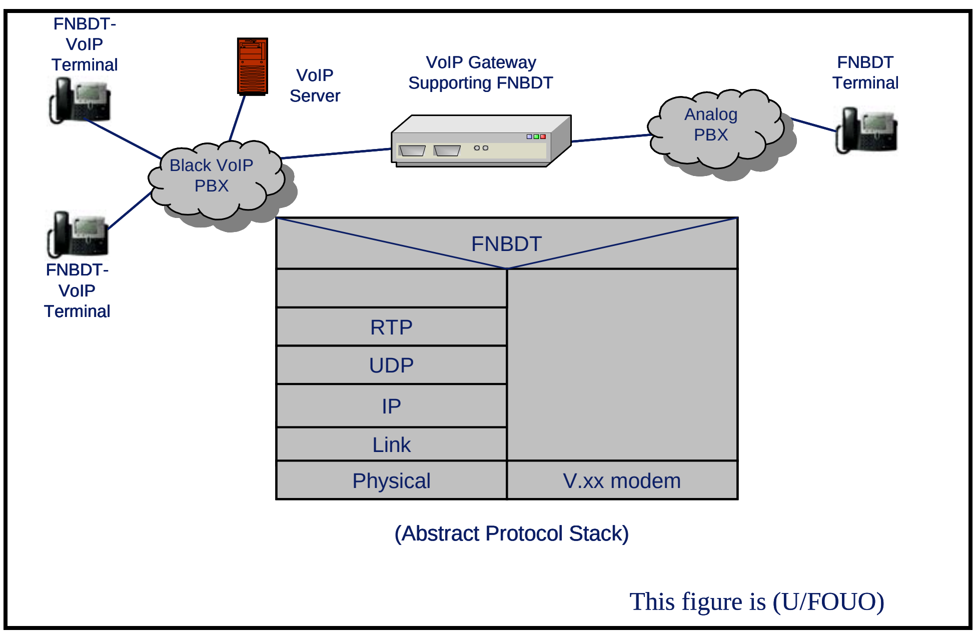


Figure 2.3-12: (U//FOUO) Secure Voice Gateway Functionality

2.3.3.2.1.2.2.2.2 (U) Advantages

(U) Gateway technology allows interoperation in at least two areas that would not be possible without gateways:

- (U) Operation with legacy equipment on circuit-switched networks
- (U) Operation with different technologies within the same user environment

2.3.3.2.1.2.2.2.3 (U) Risks/Threats/Attacks

(U//FOUO) The basic risk associated with Black gateway technology is DoS. If an adversary can gain access to the control mechanisms of the gateway, traffic channels can be blocked such that users can be kept from using them. There is no additional risk associated with the confidentiality of the user data since it is not decrypted at the gateway.

2.3.3.2.1.2.2.3 (U) Maturity

(U) Commercial signaling and media gateways are Mature (TRLs 7 – 9) and exist for solving specific problems within specific bounds. For instance, the gateway technology associated with interoperating standard non-secure calls between VoIP systems and the PSTN is well understood and has been implemented in many forms.

(U//FOUO) However, the non-standard variations required for secure voice systems are Emerging (TRLs 4 – 6). Commercial vendors have not seen a business case for defining and implementing gateways containing modem functionality as will be required for secure voice interoperation.

(U//FOUO) Commercial VoIP systems on system-high networks are Mature (TRL 9 - successful mission operations). Secure Voice variants are Emerging (TRL 5 -breadboard evaluation in relevant environment).

2.3.3.2.1.2.2.4 (U) Standards

(U) The following standards are used for gateway control in VoIP systems:

- (U) MEGACO, also referenced as Gateway Control Protocol (GCP). RFC 3525, formerly RFC 3015. Also published by the ITU-T as Recommendation H.248.1
- (U) Media Gateway Control Protocol (MGCP), RFC 3435.

2.3.3.2.1.2.2.5 (U) Cost/Limitations

(U//FOUO) Use of commercial media gateways is a cost-effective approach for VoIP systems that provide security by residing on system-high networks. For VoIP systems that require FNBDT security there are at least two options to provide the necessary gateways:

- (U//FOUO) Dedicated special-purpose gateway that leaves out the transcoder function and includes the modem function
- (U//FOUO) Modifications to commercial gateways to allow a client to bypass the transcoder in the gateway and route the information through a modem instead

(U//FOUO) Either of these options will result in additional complexity and associated cost.

2.3.3.2.1.2.2.6 (U) Dependencies

(U//FOUO) Gateway technology is highly dependent on the specific systems a particular gateway is providing interoperability between. A gateway is designed to be completely compatible with a particular system on each side. If a third system is introduced into the architecture, it is highly likely that a separate gateway will be required.

2.3.3.2.1.2.3 Secure Voice over IP

(U//FOUO) Secure VoIP technologies described here secure the voice bearer, or user voice packets. Secure VoIP call or session control used to establish calls is addressed in section 2.3.3.2.2.2.1.

2.3.3.2.1.2.3.1 (U) Technical Detail

(U//FOUO) Security technologies considered for VoIP voice packets include:

- (U) Secure Real-Time Protocol (SRTP)
- (U//FOUO) FNBDT over RTP
- (U//FOUO) Secured voice, such as FNBDT, over V.150 Modem Relay Simple Packet Relay Transport protocol (SPRT).

(U//FOUO) A brief introduction to the VoIP technology is presented before a description of potential security technologies. (VoIP call control is described in section 2.3.3.2.2.2.1.) VoIP architectures typically include control planes to set up VoIP calls and execute network services. They also include bearer planes used to transfer voice packets between users after the call has been established. H.323 and SIP are the leading protocol systems used for VoIP call control. Other notable VoIP protocols, specifically MGCP and GCP/H.248/MEGACO reside between control and bearer planes. They are used when VoIP-PSTN Gateways and Multimedia conference units are decomposed into Media Gateway Controllers (MGCs) and Media Gateways (MGs). MGCs use protocols such as MGCP to control the bearer path through MGWs.

(U//FOUO) QoS protocols and systems, such as RSVP and DiffServ, are complementary technologies needed to support VoIP services, but are not call control protocols themselves. QoS should be established or negotiated outside of the voice bearer plane as part of the overall call set up process, and subsequently applied to the actual voice stream packets. Security mechanisms are needed to protect QoS mechanisms, but such are outside the scope of this section.

(U) RTP is used in all common VoIP systems to transport voice packets between users. A closer look at RTP follows.

2.3.3.2.1.2.3.1.1 (U) RTP and RCTP Overview

(U) Real-Time Transport Protocol is designed to transport real-time applications over IP networks. RTP runs in conjunction with RTCP (RealTime Control Protocol), which provides feedback to applications about the quality of the media transmission.

(U) RTP provides time-stamping and Sequence Numbering of the Multimedia packets to enable synchronization of a received media stream. As shown in Figure 2.3-13, RTP, along with RTCP, reside on UDP. Reliability mechanisms such as re-transmits are not included since latency and jitter are more important to voice quality than bit errors or occasional voice packet losses. A description of the fields within the RTP header follows the figure.

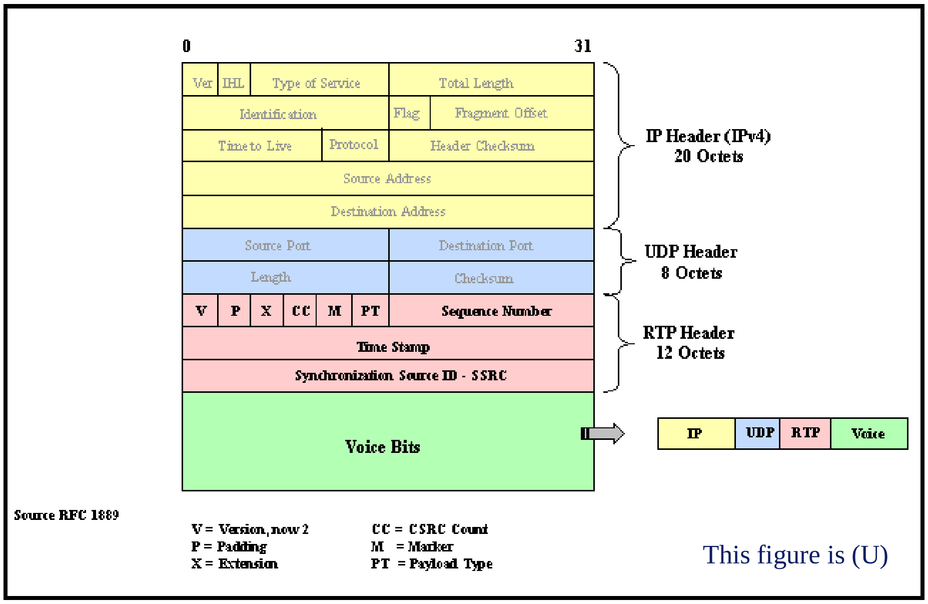


Figure 2.3-13: (U) Real-Time Protocol

(U) Version (V): 2 bits - This field identifies the version of RTP. The current version is two (2).

(U) Padding (P): 1 bit - If the padding bit is set, the packet contains one or more additional padding octets at the end that are not part of the payload. Padding may be needed by some encryption algorithms with fixed block sizes or for carrying several RTP packets in a lower-layer protocol data unit.

(U) Extension (X): 1 bit - If the extension bit is set, the fixed header is followed by exactly one header extension.

(U) CSRC count (CC): 4 bits - The (CSRC) count contains the number of CSRC identifiers that follow the fixed header. CSRCs are media contributors that reside behind a conference unit.

(U) Marker (M): 1 bit - The interpretation of the marker is defined by a profile. It is intended to allow significant events such as frame boundaries to be marked in the packet stream. A profile may define additional marker bits or specify that there is no marker bit by changing the number of bits in the payload type field.

(U) Payload type (PT): 7 bits This field identifies the format of the RTP payload and determines its interpretation by the application. A profile specifies a default static mapping of payload type codes to payload formats. An RTP sender emits a single RTP payload type at any given time; this field is not intended for multiplexing separate media streams.

(U) Sequence number: 16 bits - The sequence number increments by one for each RTP data packet sent. This number can be used by the receiver to detect packet loss and to restore packet sequence. The initial value of the sequence number is random (unpredictable) to make known-plaintext attacks on encryption more difficult, even if the source itself does not encrypt, because the packets may flow through a translator that does.

(U) Time-stamp: 32 bits - The time-stamp reflects the sampling instant of the first octet in the RTP data packet. The sampling instant must be derived from a clock that increments monotonically and linearly in time to allow synchronization and jitter calculations.

(U) SSRC: 32 bits - The Synchronization Source Real-time Content (SSRC) field identifies the synchronization source. This identifier is chosen randomly, with the intent that no two synchronization sources within the same RTP session will have the same SSRC identifier.

(U) CSRC list: 0 to 15 items, 32 bits each - The Contributing Source Real-time Content (CSRC) list identifies the contributing sources for the payload contained in this packet.

(U) RTCP runs in conjunction with RTP. Receiving participants send periodic information, using RTCP, back to the originating source to convey quality information about the received data. RTCP provides the following services:

- (U) Quality Monitoring and Congestion Control – This is the primary function of RTCP, and it provides feedback to senders about the quality of the connection. The sender can use this information to adjust transmission. Also, third party monitors can use the information to access network operation.
- (U) Source Identification – The source field in the RTP header is a 32 bit identifier, randomly generated, and not user friendly. RTCP provides a more user friendly identification of the 32 bit RTP source field by providing a global identification of session participants. This information is typically, user name, telephone number, email address, etc.
- (U) Inter-Media Synchronization – RTCP sends information that can be used to synchronize audio and video packets.
- (U) Control Information Scaling – As the number of participants increase, the amount of control information must be scaled down to allow sufficient bandwidth for the RTP channels. This is done by the RTP protocol by adjusting the RTCP generation rate. Typically, the control bandwidth is limited to 5% of the RTP traffic.

(U//FOUO) Each RTCP packet begins with a fixed part similar to that of RTP data packets, followed by structured elements that may be of variable length according to the packet type. The alignment requirement and a length field in the fixed part are included to make RTCP packets stackable. Multiple RTCP packets may be concatenated without any intervening separators to form a compound RTCP packet that is sent in a single packet of the lower layer protocol, such as UDP. There is no explicit count of individual RTCP packets in the compound packet since the lower layer protocols are expected to provide an overall length to determine the end of the compound packet.

(U//FOUO) Figure 2.3-14 displays the format of one of the five RTCP messages—the RTCP Send Report. RTP receivers provide reception quality feedback using RTCP report packets which may take one of two forms depending upon whether or not the receiver is also a sender. The only difference between the sender report (SR) and receiver report (RR) forms, besides the packet type code, is the sender report includes a 20-byte sender information section active senders can use.

(U//FOUO) It is expected that reception quality feedback will be useful not only for the sender but also for other receivers and third-party monitors. The sender might modify its transmissions based on the feedback. Receivers can determine whether problems are local, regional or global. Network managers can use profile-independent monitors that receive only the RTCP packets and not the corresponding RTP data packets to evaluate the performance of their networks for multicast distribution.

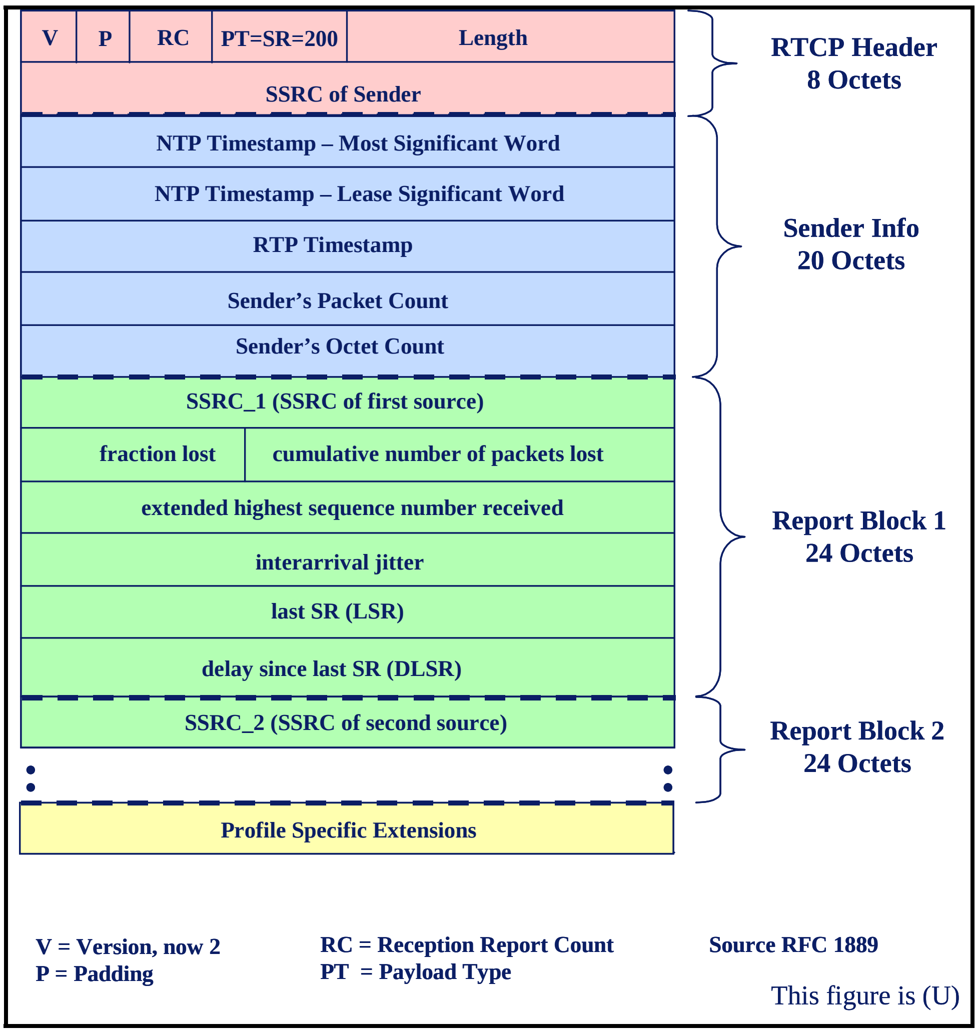


Figure 2.3-14: (U) RTCP Sender Report Format- Sender Report (SR)

(U) Secure RTP (SRTP) and Secure RTCP (SRTCP) per RFC 3711

(U//FOUO) SRTP/SRTCP are used to protect the RTP/RTCP protocols. SRTP supports both IP unicast and multicast communications. SRTP is a commercial security system and no type 1 versions are available. Therefore, SRTP can be used within a security domain, but without further development is not advised to use to secure voice traffic across separate security environments. Therefore, another lower layer protocol, such as IPsec, should be used to transport secure voice across security domains.

(U//FOUO) Secure RTP is used to authenticate and protect RTP headers and payloads. It is defined as an extension to the RTP Audio/Video profile per RFC 3551. Each SRTP stream is organized around cryptographic contexts that senders and receivers use to maintain cryptographic state information. The cryptographic context is uniquely mapped to the combination of:

- (U) The destination IP address
- (U) The destination port
- (U) The SSRC (as seen in the RTP header).

(U//FOUO) SRTP session keys are cryptographically derived per the RFC from master keys and salt keys that are initialized via key management. The salt keys are updated per the RFC for use in subsequent session key derivations. The session keys are then applied to the voice media stream to provide encrypted voice.

(U//FOUO) SRTP does not define or mandate a specific key management protocol. However, it does place requirements and considerations on the key management system. These considerations are described in the dependencies section below.

<u>(U) The SRTP Protocol</u>

(U//FOUO) Figure 2.3-15 illustrates the format of SRTP.

(U//FOUO) As can be seen from Figure 2.3-15, the SRTP format uses the RTP header and RTP payload, followed by the SRTP MKI and Authentication tag fields. The entire SRTP packet is authenticated while the RTP payload and SRTP MKI and authentication tags are encrypted.

(U//FOUO) The optional SRTP MKI (Master Key Identifier) field identifies the master key used in the session. It does not indicate cryptographic context. The MKI is defined and distributed by the key management system.

(U//FOUO) The SRTP authentication tag is used to carry message authentication data. The tag provides authentication of the RTP header and payload and indirectly provides replay protection by authenticating the RTP sequence number. The tag is recommended for use by the RFC.

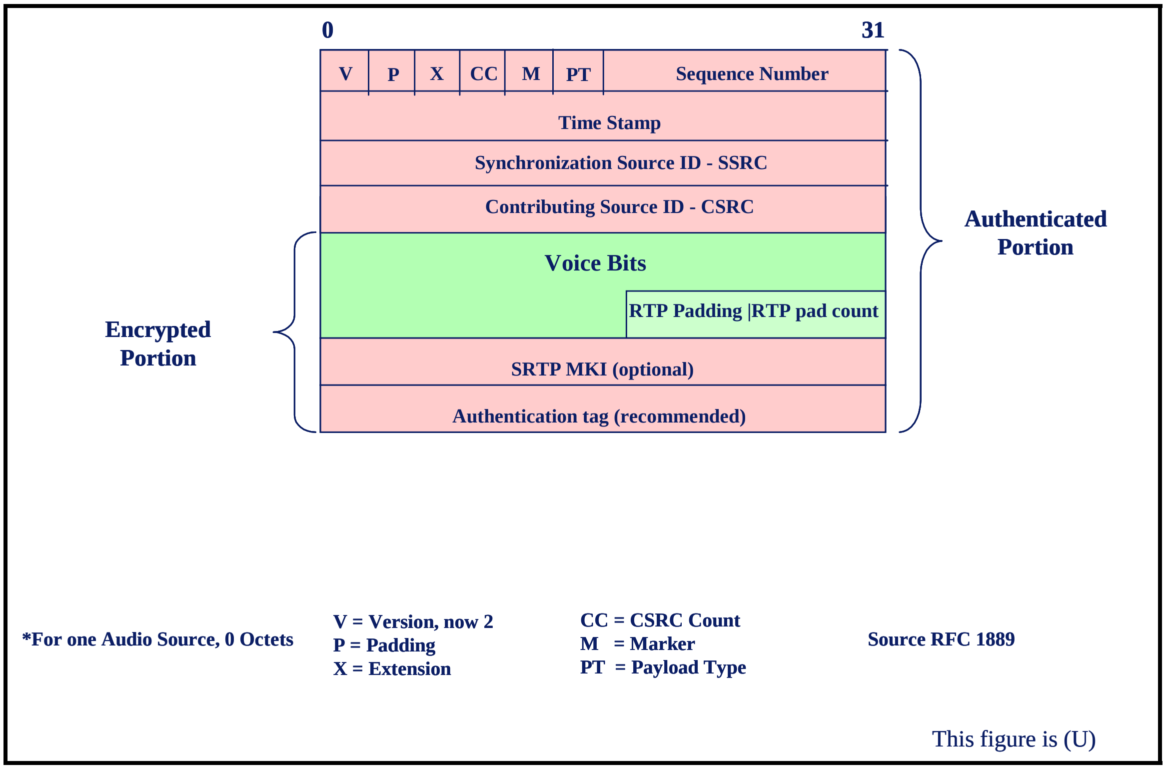


Figure 2.3-15: (U) SRTP Format

<u>(U) The SRTCP protocol:</u>

(U//FOUO) Figure 2.3-16 illustrates the format of SRTCP:

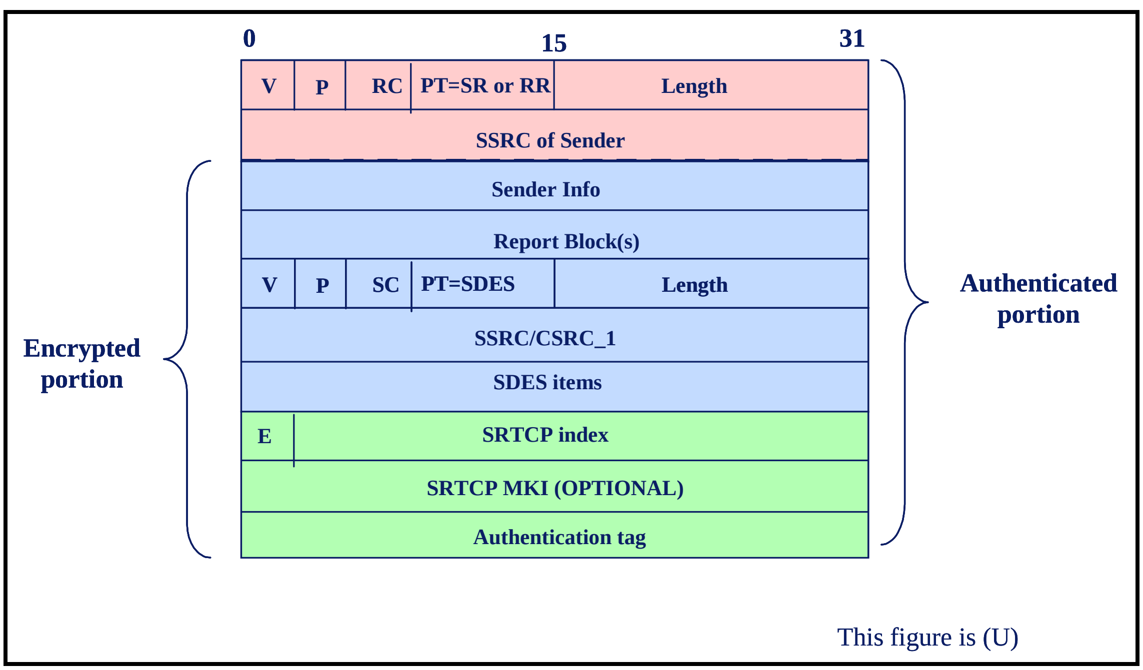


Figure 2.3-16: (U) SRTCP Format

(U//FOUO) As can be seen from Figure 2.3-16, the SRTCP format uses the RTCP header and RTCP payload reports, followed by the SRTP 'E', SRTCP index, SRTPC MKI, and Authentication tag fields. The entire SRTCP packet is authenticated while the RTCP payload is encrypted along with SRTCP specific fields. (Note that Figure 2.3-16 shows a generalized representation of the RTCP reports, but this does impact the discussion of SRTCP fields.)

(U//FOUO) The 'E' field is a one-bit flag that indicates if the current RTCP packet is encrypted.

(U//FOUO) The SRTCP index is a 31-bit counter for the SRTCP packet. It is initially set to zero before the first SRTCP is sent and incremented by one after each SRTCP packet. It is not reset to zero after a rekey event to help provide replay protection.

(U//FOUO) The optional SRTCP MKI field indicates the master key used to derive the session key for the RTCP context.

(U//FOUO) The SRTCP authentication tag is used to carry message authentication data. The tag provides authentication of the RTCP header and payload. The tag is recommended for use by the RFC.

<u>(U) Encryption algorithms</u>

(U//FOUO) SRTP calls out AES in counter mode for encryption and HMAC-SHA1 for message authentication and integrity. The RFC explicitly states that any transforms added to SRTP must be added with a companion standard track RFC that exactly defines how the transform is used with SRTP.

<u>(U) FNBDT over RTP</u>

(U//FOUO) The second technology considered for secure voice is to create an RTP payload type for FNBDT type 1 secured voice. Please refer to section 2.3.3.2.1.2.1 for a description of FNBDT. The new RTP profile is defined to support both FNBDT signaling and data. This FNBDT media type needs to be identified and negotiated between clients within the GIG VoIP call control architecture. The RTP protocol field described above contains a GIG unique payload value indicating FNBDT to GIG users. In such a scenario, a client could start a clear voice call using an Internet Assigned Numbers Authority (IANA) standard RTP payload type (voice codec) and then use call control signaling to transition to the FNBDT profile.

(U//FOUO) Note that certain FNBDT modes add overhead to the clear voice codec approaches in order to maintain crypto-synchronization. Differences in network resource requirements when transitioning between clear and secured FNBDT voice would need to be accounted for in the GIG QoS architecture.

<u>(U) Secured voice, such as FNBDT encryption, over V.150.1 Modem Relay</u>

(U//FOUO) Secure voice, such as FNBDT encryption, over V.150.1 modem relay applies type 1 secured voice over a commercial standard real-time transport mechanism for data. Please refer to section 2.3.3.2.1.2.1 for a description of FNBDT. The following is a brief overview of V.150.1 modem relay.

(U//FOUO) ITU specification V.150.1 Modem Relay, hereafter referred to to V.150.1 MR, is a VoIP-PSTN gateway feature. It is designed to allow the successful transfer of data from a modem on a circuit network, through a PSTN-VoIP GW and across an IP network, through a second VoIP-PSTN GW and back onto a second modem over circuit network. FNBDT secure voice over V.150.1 MR exploits the V.150.1 SPRT protocol as illustrated in Figure 2.3-17 below.

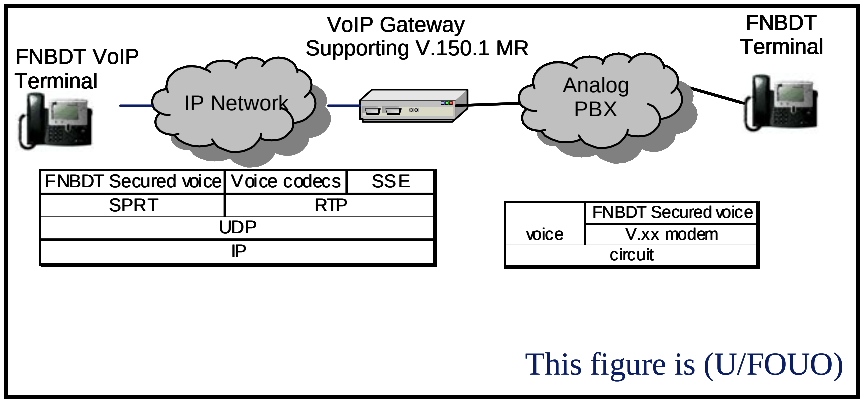


Figure 2.3-17: (U//FOUO) FNBDT over V.150.1 Modem Relay

(U//FOUO) V.150.1 supports several modes that allow a user to multiplex between voice and data transport. As can be seen from Figure 2.3-17, V.150.1 enables clear voice and the V.150.1 defined State Signaling Protocol to be carried over RTP. SSE is used to transition between voice and modem data transport modes at a V.150.1 PSTN-VoIP GW. As such, State Signaling Event (SSE) instructs the GW to initiates a V.xx modem on the circuit network in preparation of making a voice to data transition. Data bearer, however, is carried over the IP network with the V.150.1 Simple Packet Relay Transport protocol, SPRT. SPRT is placed on UDP and includes both reliable and transparent sequenced modes. FNBDT secured voice is carried over the SPRT transparent sequenced mode, which is designed to support real-time data. SPRT includes sequence numbers to facilitate proper packet ordering at receivers. As such, V.150.1 MR is able to transport type 1 secured voice between VoIP and circuit networks using a V.150.1 commercial standard VoIP GW.

(U) Figure 2.3-18 below illustrates the SPRT format:

1	2	3 4 5 6 7	8	9 10 11 12 13 14 15	16 17	18 19 20 21 22 23 24 25 26 27 28 29 30 31
X	SSID		R	PT	TC	Sequence Number
NOA		Base Sequence Number			TCN	Sequence Number
TCN		Sequence Number			TCN	Sequence Number

This figure is (U)

Figure 2.3-18: (U) V.150.1 Simple Packet Relay Transport for IP networks

(U) The SPRT header fields are summarized as follows:

(U) X = Header Extension Bit, currently reserved by ITU

(U) SSID = SubSession ID, used to identify a media stream

(U) R = reserved by ITU

(U) PT = payload type. The payload is set to the value assigned by the media stream by call control signaling. Note that the R and PT field together are consistent with the same fields seen in the RTP header such that clients and GWs can transition between voice/RTP and data/SPRT over the same UDP port.

(U//FOUO) TC – Transport Channel number. FNBDT uses transport channel 3, designed for real-time data without acknowledgements.

(U//FOUO) The sequence number field is incremented with each SPRT packet, similar to the sequence number in RTP.

(U//FOUO) NOA – Number of Acknowledgments. Acknowledgments are set to zero for FNBDT and other real-time data services.

(U//FOUO) The Base Sequence number field is used to manage re-transmits. It is set to zero for TC= 3, the channel used by FNBDT.

(U//FOUO) TCN and subsequent sequence number fields are used for re-transmits. These fields are not used with FNBDT over V.150.1 MR.

(U//FOUO) In summary, an FNBDT over V.150.1 client can first establish a clear voice call and send clear voice via RTP. It can then use SSE to transition to data mode. Once in data mode, the client then used SPRT to transport FNBDT signaling and secured voice.

2.3.3.2.1.2.3.2 (U) Usage Considerations

2.3.3.2.1.2.3.2.1 (U) Implementation Issues

<u>(U) SRTP</u>

(U//FOUO) Each media stream in a multimedia session requires its own SRTP session key. This multiplies the potential number of security contexts initiated per user. This is a concern for mobile multimedia services with limited battery and processing power. More security contexts could also multiply the amount of key management traffic.

(U//FOUO) Forward error correction and interleaving, if required by the RTP media type, need to be completed before application of SRTP.

(U//FOUO) SRTP cannot span into non-IP networks, such as the PSTN or DSN. Therefore, a VoIP-PSTN GW would need to terminate SRTP and invoke another security mechanism for the PSTN side of a VoIP to PSTN call. This requires a complex, Red GW function.

(U//FOUO) Note that SRTP can be used for end-to-end security in half duplex voice conferences using multicast. But full duplex conferences require a Red conference unit, either at each client or in a central server.

<u>(U) FNBDT over RTP</u>

(U//FOUO) The custom RTP profile developed for RTP complicates the use of existing VoIP call control mechanisms, which will need to be extended for this unique media type. It should also be noted that the RTP header time stamp is not used as originally intended since the RTP payload contains a mixture of FNBDT security signaling besides ciphered voice. FNBDT over RTP does not fit within conventional VoIP-circuit GWs. A custom GW would be required for FNBDT over RTP. See section 2.3.3.2.1.2.2 for further discussion of GW topics.

<u>(U) FNBDT over V.150.1 Modem Relay</u>

(U//FOUO) V.150.1 MR is defined for PSTN-IP-PSTN scenarios and as such is a GW architectural element. Therefore, this GW function would need to be collapsed into a voice client for application in an all IP environment. This may be considered complex for a mobile user device. Since V.150.1 MR is not a widely used transport mechanism in IP networks, it potentially introduces a new network transport mechanism in the GIG specifically for secure voice.

2.3.3.2.1.2.3.2.2 (U) Advantages

(U//FOUO) SRTP fits well within conventional VoIP architectures and promises to become a widely known commercial standard. It is less complex than other approaches when used in an all IP network of a single-security domain.

(U//FOUO) FNBDT is a proven type 1 protocol. The use of RTP fits within conventional VoIP architectures, although it is extended for the non-standard FNBDT media type. But FNBDT over RTP would require custom black circuit-VoIP GWs.

(U//FOUO) V.150.1 MR can be used within an all IP network as well across black V.150.1 PSTN GWs to legacy FNBDT devices. As such, it offers the potential to provide type 1 security from a VoIP to a PSTN-based terminal as it leverages an established type 1 security protocol.

2.3.3.2.1.2.3.2.3 (U) Risks/Threats/Attacks

<u>(U) SRTP</u>

(U//FOUO) SRTP does support type 1 algorithms without extending the protocol with a new standards track RFC. As such, it should not be used to transport secure voice across security domain boundaries.

(U//FOUO) RTP uses the 16-bit RTP header sequence number to help set the KG state. Use of the RTP sequence number to set KG state may not be sufficient for type 1 algorithms. The RTP header used is subject to manipulation, although the SHA-1 authentication mechanism provides at least partial protection.

(U//FOUO) SRTP would also require custom, red VoIP- circuit GWs. Since SRTP requires Red GWs to reach circuit networks, it may not be the leading security protocol candidate for secure voice.

<u>(U) FNBDT over RTP</u>

(U//FOUO) There are no risks, threats, or attacks unique to FNBDT over RTP identified that are not common to any type 1 application transferred over an RTP/UDP/IP stack. Specifically, the IP, UDP, and RTP headers are not protected nor authenticated.

<u>(U) FNBDT over V.150</u>

(U//FOUO) There are no risks, threats or attacks unique to FNBDT over V.150.1 identified that are not common to any type 1 application transferred over a SPRT/UDP/IP stack. Specifically, the IP, UDP and SPRT headers are not protected nor authenticated.

2.3.3.2.1.2.3.3 (U) Maturity

<u>(U) SRTP</u>

(U//FOUO) SRTP is Emerging with an estimated TRL of 4 since it is well specified and released as an RFC, dated March 2004. It is assumed that portions of the function have been demonstrated with experimental code by SRTP within the IETF community. SPRT products are widely available in commercial products.

(U//FOUO) Areas of further study and development are recommended before SRTP can be used within the GIG—specifically:

- (U//FOUO) A Key management system that incorporates SRTP requirements needs to be defined and developed
- (U//FOUO) A concept of operations that describes how SRTP is supported within the GIG call/session control architecture needs to be developed
- (U//FOUO) Methods to transition between clear and secure voice within a single session using SRTP need to be defined and developed
- (U//FOUO) An evaluation of how SRTP might evolve to support type 1 security might be considered
- (U//FOUO) Performance evaluation and prediction of SRTP within mobile environments should be addressed.

(U) FNBDT over RTP

(U//FOUO) FNBDT and RTP are both Mature with a TRL of 9, since they have been proven in multiple product deployments. But use of an FNBDT RTP profile or media type is new and has progressed little past laboratory demonstration. As such, we consider the combination of FNBDT and RTP to be Emerging with a TRL of 4.

(U//FOUO) Areas of further study and development are recommended before FNBDT over RTP can be used within the GIG—specifically:

- (U//FOUO) FNBDT rekey over IP needs to be developed
- (U//FOUO) A concept of operations that describes how FNBDT over RTP is supported within the GIG call/session control architecture needs to be developed
- (U//FOUO) Methods to transition between clear and secure voice within a single session using FNBDT over RTP need to be defined and developed
- (U//FOUO) FNBDT over RTP is currently defined for point-to-point communications. An evaluation of how FNBDT and RTP constructs could be extended to support multicast communications is advised
- (U//FOUO) Performance evaluation and prediction of FNBDT over RTP within mobile environments should be addressed

(U) FNBDT over V.150.1

(U//FOUO) FNBDT is a Mature secure technology as merits a TRL of 9.

(U//FOUO) V.150.1 MR is a new, immature transport technology that may not be widely used or supported in the commercial world. Several sections in the ITU are labeled, ‘to be defined,’ and standard evolution is likely. Even so, a commercial manufacturer has demonstrated V.150.1 MR capability and is likely to offer the function in commercial products. For this reason, FNBDT over V.150.1 is Emerging (TRL 4).

(U//FOUO) Areas of further study and development are recommended before V.150.1 MR can be used within the GIG—specifically:

- (U//FOUO) FNBDT rekey over IP needs to be defined and developed
- (U//FOUO) A concept of operations that describes how V.150.1 media type is supported within the GIG call/session control architecture needs to be developed
- (U//FOUO) Methods to transition between clear and secure voice within a single session using RTP – V.150.1 (SPRT) session need to be defined and developed
- (U//FOUO) FNBDT over V.150.1 is currently defined for point-to-point communications. An evaluation is advised on how FNBDT and V.150.1 constructs could be extended to support multicast communications

- (U//FOUO) Performance evaluation and prediction of V.150.1 within mobile environments should be addressed.

2.3.3.2.1.2.3.4 (U) Standards

Table 2.3-5: (U) Secure Voice over IP Standards

This table is (U//FOUO)

Name	Description
FNBDT-210	Signaling Plan Revision 2.0
ITU V.150	Procedures for the end-to-end connection of V-series DCEs over and IP network
RFC 3550	RTP: A Transport Protocol for Real-Time Applications
RFC 3711	The Secure Real-time Transport Protocol (SRTP)

This table is (U//FOUO)

2.3.3.2.1.2.3.5 (U) Cost/Limitations

(U//FOUO) SRTP cannot be used with COTS PSTN GWs to reach secure voice devices on the PSTN or DSN. SRTP must be terminated at the GW. This lack of PSTN interoperability: (1) can complicate migration plans, (2) might restrict mobile GIG user communications in less developed countries and (3) can restrict secure voice with less developed coalition partners. A custom Red PSTN GW is required.

(U//FOUO) FNBDT over RTP has the same restriction as SRTP. It cannot be used with COTS PTSN GWs to reach secure voice devices on the PSTN or DSN. A custom Black PSTN GW is required. Furthermore, FNBDT currently does not support multicast or groups call. FNBDT standards development is required for group calling.

(U//FOUO) V.150.1 may not be widely used in the commercial market. It might be used exclusively for secure voice within the GIG. As such, it is a network transport mechanism that may not enjoy economies of scale as other approaches might. Furthermore, V.150.1 was not defined for multicast groups, so a concept of operations for secure group calls utilizing V.150.1 needs to be developed.

(U//FOUO) FNBDT currently does not support multicast, or group calls. Further FNBDT standards development is required.

2.3.3.2.1.2.3.6 (U) Dependencies

<u>(U) SRTP</u>

(U) Key Management Dependencies and interaction

(U//FOUO) SRTP places a number of dependencies upon key management. Section 8.2 of the RFC details a list of parameters the key management system provides including:

- (U//FOUO) Master key parameters for an SSRC

- (U//FOUO) Salt keys parameters for an SSRC

- (U//FOUO) Initial RTP sequence number and other crypto context index parameters (optional)

(U//FOUO) Clients will need to account for the amount of traffic protected with a single master key and request a rekey from the key management system based on specific usage criteria. The key management system can, of course, push keys to SRTP clients.

<u>(U) QoS management</u>

(U//FOUO) Network QoS mechanisms suitable for VoIP and RTP should be sufficient for SRTP.

(U) FNBDT over RTP and FNBDT over V.150.1 MR

(U//FOUO) FNBDT has specific key management requirements and specifications and currently supported with deployed facilities. These facilities may need to be upgraded to meet GIG needs.

<u>(U) QoS management</u>

(U//FOUO) Network QoS mechanisms need to be developed that take into account the resource utilization of FNBDT, particularly between clear and secure voice transitions.

2.3.3.2.1.2.3.7 (U) Alternatives

(U//FOUO) IP layer security could be used as an alternative to SRTP and FNBDT over RTP.

2.3.3.2.1.2.3.8 (U) Complementary Techniques

(U//FOUO) Type 1 IPsec is needed to secure SRTP protected voice traffic across security domains.

2.3.3.2.2 (U) Transport & Network Layer Technologies

2.3.3.2.2.1 (U) Non-Real-Time Data Technologies

2.3.3.2.2.1.1 (U) IP Layer Security

(U//FOUO) IP layer security enables the Black Core concept allowing IP layer routing and sub-network layer switching to occur on the Black Core, whereas traditional link security required all routers and switches to be Red.

2.3.3.2.2.1.1.1 (U) Technical Detail

(U//FOUO) Commercial IPsec can be used to protect SBU traffic, and HAIPE can be used to protect classified traffic. HAIPE was originally based on the existing commercial IPsec standards, but has shifted from these standards to provide the higher level of security necessary in a Type-1 application.

(U//FOUO) Commercial IPsec defines separate protocols for confidentiality and authentication, but the confidentiality protocol (ESP) provides an optional authentication mechanism. HAIPE requires authentication as well as confidentiality.

(U//FOUO) Commercial IPsec has defined both a transport mode and a tunnel mode and is intended to support both end system and intermediate system implementations. HAIPE has defined only a tunnel mode and is intended to support only intermediate system (INE) implementations. The GIG vision is to migrate HAIPE back to end system implementations, and consequently some modifications will be required to HAIPE to achieve this vision.

(U//FOUO) HAIPE v1 supported IPv4 only, and HAIPE v2 is intended to support both IPv4 and IPv6. The GIG vision is to migrate to IPv6.

(U//FOUO) HAIPE supports a Security Policy Database (SPD) to control the flow of IP datagrams. HAIPE supports selectors such as source/destination addresses (IPv4 and IPv6) to map IP datagram traffic to policy in the SPD. Each entry specifies the relevant selectors and whether data should be tunnel-mode encrypted or discarded. If an SPD entry cannot be found for an IP datagram, the IP datagram is discarded. Entries in the SPD are ordered. The SPD can be managed locally by the administrator/operator HMI or remotely from the SMW.

(U//FOUO) HAIPE also supports a Security Association Database (SAD). The SPD is consulted in formation of SA entries in the SAD during an Internet Key Exchange (IKE). Separate distinct SAs are used for inbound and outbound traffic. The two SAs use the same Traffic Encryption Key (TEK), but have different SPI values. Entries in the SAD are not ordered. The SAD is consulted in the processing of all traffic including non-IPsec traffic (i.e., bypassed/regenerated traffic as well as traffic encrypted in tunnel mode).

(U//FOUO) HAIPE utilizes the ESP tunnel mode to provide data integrity, anti-replay protection, confidentiality, and authentication. The original Red IP datagram is encapsulated with the HAIPE ESP protocol and then a Black IP protocol, as shown in Table 2.3-6. Table 2.3-6 indicates a total overhead of 83 octets (or 664 bits) for each Red datagram (assuming Black IP is v4). The HAIPE trailer padding includes both crypto padding and TFS padding. Crypto padding varies from 0-47 octets (HAIPE supports crypto block sizes of 4, 8, and 48 octets), and an average value of 23 octets is assumed for the overhead calculation in Table 2.3-6. No TFS padding is assumed in the overhead calculation in Table 2.3-6. Of course, the addition of TFS padding would increase the overhead.

Table 2.3-6: (U//FOUO) HAIPE ESP Tunnel Mode Encapsulation

This table is (U//FOUO)

Field		Authenticated	Encrypted	Overhead	
				Bits	Octets
Black IP Header				160	20
HAIPE ESP Header	SPI	X		32	4
	ESP Sequence Number			32	4
	State Variable			128	16
	Payload Sequence Number	X	X	64	8
Red IP Datagram	Red IP Header	X	X	-	-
	Red IP Payload	X	X	-	-
HAIPE ESP Trailer	Padding (Crypto + TFS)	X	X	184	23
	Dummy	X	X	8	1
	Pad Length	X	X	16	2
	Next Header	X	X	8	1
	Authentication Data		X	32	4
Total				664	83

This table is (U//FOUO)

(U//FOUO) Note that HAIPE provides authentication (anti-spoof protection) of the Red IP datagram and parts of the HAIPE header and trailer as indicated in the Authenticated column in Table 2.3-6. PDUs that fail the authentication check are discarded. This may be undesirable for voice and video data where a few bit errors are tolerable. HAIPE provides confidentiality (encryption) of the Red IP datagram and parts of the HAIPE header and trailer as indicated in the Encrypted column in Table 2.3-6.

(U//FOUO) The 32-bit SPI identifies the security association to the receiving HAIPE device. The SPI is either calculated from key material and peer information (for PPKs) or developed during the IKE exchange (for automatic TEK generation).

(U//FOUO) HAIPE uses the payload Payload Sequence Number (PSEQN) for anti-replay service. Therefore even though transmitting HAIPE devices initially set the ESP SEQN value to a random number and increment for each packet set, receiving HAIPE devices ignore the ESP SEQN value.

(U//FOUO) The state variable is used to synchronize the crypto state of the transmitting and receiving HAIPE device, and does not repeat for any given TEK. The state variable is transmitted with each PDU so that the receiving HAIPE device can independently decrypt each PDU. Table 2.3-7 shows contents of the state variable.

Table 2.3-7: (U//FOUO) HAIPE State Variable Content

<table>
<tr><td colspan="4">This table is (U//FOUO)</td></tr>
<tr><th>Field</th><th>Bits</th><th>Value On Wire</th><th>Encryption/ Decryption Value</th></tr>
<tr><td>Update Count</td><td>16</td><td colspan="2">Indicates daily update count of TEK</td></tr>
<tr><td>Unique</td><td>69</td><td colspan="2">Unique</td></tr>
<tr><td>LRS</td><td>36</td><td rowspan="3">Unique</td><td>Stepped when SEG# has value 0</td></tr>
<tr><td>SEQ#</td><td>4</td><td>All zeros</td></tr>
<tr><td>SEG#</td><td>3</td><td>Stepped from 0-3 for WEASLE Mode</td></tr>
<tr><td>Total</td><td>128</td><td>-</td><td>-</td></tr>
<tr><td colspan="4">This table is (U//FOUO)</td></tr>
</table>

(U//FOUO) The update count field represents the number of daily updates performed on the TEK. The receiver uses to determine which update version of the TEK was used by the transmitter to encrypt the PDU.

(U//FOUO) The Unique, LRS, SEQ#, and SEG# fields are unique on the wire for each PDU.

(U//FOUO) The initial value for the Linear Recursive Sequence (LRS) is transmitted on the wire and is uniquely generated for each PDU. During encryption or decryption processing of crypto blocks of the same PDU, the LRS is stepped each time the SEG# has the value zero illustrated in Figure 2.3-19. The polynomial for the LRS is $1+x^{11}+x^{36}$.

(U//FOUO) The SEQ# field is unique on the wire, but all zeros for encryption and decryption.

(U//FOUO) The SEG# is unique on the wire. During encryption or decryption processing of crypto blocks of the same PDU, the SEG# is stepped from 0 to 3 in WEASEL mode as illustrated in Figure 2.3-19.

(U//FOUO) The PSEQN value provides anti-replay services for HAIPE. The PSEQN value is both authenticated and encrypted. The PSEQN value is initialized to zero by the transmitting HAIPE upon SA setup and incremented for each PDU sent for the duration of the TEK. The receiving HAIPE uses the PSEQN value to detect and discard PDUs that are replayed.

(U//FOUO) Inner IP header fields are coded in accordance with RFC 2401.

(U//FOUO) Padding ensures the encrypted PDU is an integer multiple of the encryption block size, which may be negotiated during the IKE. Padding is also used to provide TFS protection. The ESP padding is added by the transmitting HAIPE and removed by the receiving HAIPE.

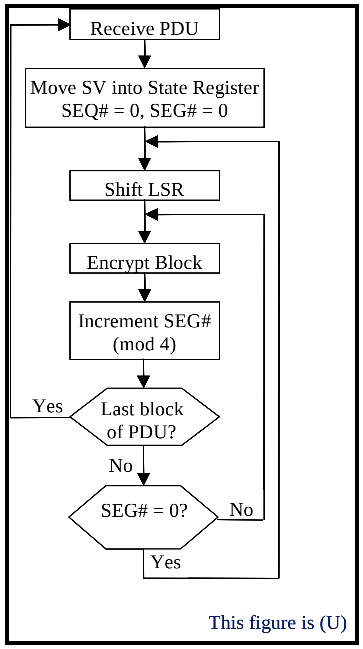


Figure 2.3-19: (U//FOUO) State Variable Stepping

(U//FOUO) The ESP dummy field is used to support TFS protection. A value of all 0s indicates a dummy PDU, whereas a value of all 1s indicates a Valid PDU.

(U//FOUO) The ESP pad length field is used by the receiving HAIPE to determine the amount of padding added by the transmitting HAIPE, so the receiver can remove this padding before forwarding the decrypted PDU to the receiving host.

(U//FOUO) The ESP next header field indicates the encapsulated protocol. In the case of tunneled user traffic, this field will indicate IPv4 (or IPv6).

(U//FOUO) HAIPE supports both the BIP-32 and SHA-1 algorithms for authentication. In either case, the authentication value is encrypted under the negotiated cryptographic algorithm operating in WEASEL mode.

(U//FOUO) The BIP-32 algorithm is a 32-bit exclusive-or function of each of the 32-bit words of data to be authenticated and a 32-bit word of hexadecimal “A”s (0xAAAAAAAA).

2.3.3.2.2.1.1.2 (U) Usage Considerations

2.3.3.2.2.1.1.2.1 (U) Implementation Issues

(U//FOUO) The application of IPsec to protect Unclassified traffic in the GIG introduces key management issues. In order to provide sufficient protection secure Type 3 key material must be generated, distributed, stored, updated and destroyed. The current KMI is only designed for Type 1 key material. Given the volume of Unclassified data in the GIG, the number of IPsec devices that must be keyed will also be significant (residing on every client in the GIG Vision). Without a key management infrastructure for Type 3 keys, deployment of IPsec to protect Unclassified traffic may be unmanageable.

(U//FOUO) HAIPE does not support dynamic routing in a multi-homed environment (i.e. an enclave fronted by more than one HAIPE. This limitation may be overcome by placing external routers behind HAIPEs, and using IP tunneling (e.g. see RFC-2784 on Generic Routing Encapsulation) between the routers to disguise the ultimate destination from the INE. This approach requires an extra IP header, and therefore increases bit overhead across the Black Core.

(U//FOUO) An alternative is to integrate a router into the Red side of the INE and to select the SA based on the next hop instead of the ultimate destination address. This approach has less bit overhead than the external router approach, but couples the routing function with the HAIPE security function.

(U//FOUO) HAIPE does not fully support QoS mechanisms for real-time traffic like voice. HAIPE does support bypass of IPv4 Type of Service (ToS) field and IPv6 Traffic Class field, but does not support reservation protocols. For example, TSAT has proposed modifications to HAIPE to provide an RSVP proxy service where a HAIPE INE would aggregate multiple Red side RSVP requests into a single Black side RSVP request.

(U//FOUO) HAIPE does not provide true end-to-end security. Currently HAIPE is designed to support INE implementations with multiple end-systems behind an INE. Even when migrated to end-systems, HAIPE will not provide true end-to-end security for some applications. For example, e-mail is a store-and-forward application with multiple IP end-systems in the path. Likewise, secure voice through a gateway will have IP end-systems as intermediate nodes. Additional security protocols may be overlaid on top of HAIPE to provide true end-to-end security (e.g. SMIME v3 for secure e-mail and FNBDT for secure voice).

(U//FOUO) HAIPE is currently not designed for end system implementation. HAIPE version 1 and version 2 only support tunnel mode and does not support transport mode. HAIPE version 3 will support both tunnel and transport modes. Anti-tamper and TEMPEST are also significant issues for a Type-1 end-system implementation.

(U//FOUO) HAIPE discovery does not support dynamic Black-side IP addresses. Dynamic Black-side IP addresses are common in a mobile IPv4 environment. The migration to IPv6 in the future will help to resolve this issue to some extent.

(U//FOUO) HAIPE has significant complexity, and was not intended for implementation in a resource-constrained environment (e.g., memory and processing power). A profile of HAIPE may be desirable for implementation in hand-held mobile devices.

(U//FOUO) HAIPE was not intended for low bandwidth and/or high BER environments. HAIPE has significantly more bit overhead than FNBDT for protecting secure voice traffic. There have been several HAIPE-lite proposals to address this issue. These proposals do reduce the bit overhead associated with HAIPE, but are still not as efficient as FNBDT for secure voice applications. HAIPE is also not tolerant to bit errors. HAIPE provides cryptographic error extension and also implements an integrity check on every packet. A single bit error will cause the packet to be discarded. This is not necessarily desirable for applications like secure voice that can tolerate a few bit errors.

2.3.3.2.2.1.1.2.2 (U) Advantages

(U//FOUO) IP layer security supports a black core concept allowing switches and routers to exist on the black side of the crypto.

2.3.3.2.2.1.1.2.3 (U) Risks/Threats/Attacks

(U//FOUO) IP layer security is somewhat susceptible to Traffic Flow Analysis. Moving IP layer security (HAIPE) back to end systems will likely increase susceptibility to Traffic Flow Analysis. Link layer security may be used to provide Traffic Flow Security (TFS) for traffic encrypted at the IP layer.

2.3.3.2.2.1.1.3 (U) Maturity

(U//FOUO) IPsec is a Mature technology in the commercial world, but continues to evolve. Type 1 IP security standards (SDNS and HAIPE) have been around for quite some time, but HAIPE continues to evolve, and the current standard is not adequate to support the long-term GIG vision. The TRLs for the IP security technology are illustrated in Table 2.3-8 below. Table 2.3-8 is based on the HAIPE Roadmap presentation dated May 12, 2004.

Table 2.3-8: (U//FOUO) IP Security Technology Readiness Levels

<table>
<tr><td colspan="4">This table is (U//FOUO)</td></tr>
<tr><th>Specification</th><th>Core/Extension</th><th>Features</th><th>TRL</th></tr>
<tr><td>IPsec (November 1998)</td><td></td><td></td><td>9</td></tr>
<tr><td>IPsec (March 2004)</td><td></td><td></td><td>2</td></tr>
<tr><td>HAIPE v1.3.5</td><td>Core</td><td>BATON and FIREFLY, IPv4</td><td>9</td></tr>
<tr><td rowspan="2">HAIPE v2.0.0</td><td>Core</td><td>MEDLEY and Enhanced FIREFLY
IPv6, QoS, Multicast
Over the Network Management</td><td rowspan="2">2</td></tr>
<tr><td>Extension</td><td>Interim Routing
Enclave Prefix Discovery Server
Foreign Interoperability</td></tr>
</table>

This table is (U//FOUO)

Specification	Core/Extension	Features	TRL
HAIPE v3 & Beyond	Core	Bandwidth Efficiency (v3)	1
		OTNK (Beyond v3)	
		Over the Network Management Enhancements (Beyond v3)	
	Extension	Standard HAIPE MIB	
		Scalable & Efficient Routing	
		End-to-End QoS	
		Voice over Secure IP (VoSIP)	

This table is (U//FOUO)

2.3.3.2.2.1.1.4 (U) Standards

(U//FOUO) The standards applicable to IP security technology are identified in Table 2.3-9 below.

Table 2.3-9: (U//FOUO) Standards Applicable to IP Security Technology

This table is (U)

Number	Title	Version	Date
	Interoperability Specification For High Assurance Internet Protocol Encryptor (HAIPE) Devices	1.3.5	May 2004
	Interoperability Specification For High Assurance Internet Protocol Encryptor (HAIPE) Devices	2.0.0	May 2004
RFC-2401	Security Architecture for the Internet Protocol http://www.ietf.org/rfc/rfc2401.txt		November 1998
	Security Architecture for the Internet Protocol http://www.ietf.org/internet-drafts/draft-ietf-ipsec-rfc2401bis-02.txt		April 2004
RFC-2402	IP Authentication Header http://www.ietf.org/rfc/rfc2402.txt		November 1998
	IP Authentication Header http://www.ietf.org/internet-drafts/draft-ietf-ipsec-rfc2402bis-07.txt		March 2004
RFC-2406	IP Encapsulating Security Payload (ESP) http://www.ietf.org/rfc/rfc2406.txt		November 1998
	IP Encapsulating Security Payload (ESP) http://www.ietf.org/internet-drafts/draft-ietf-ipsec-esp-v3-08.txt)		March 2004

This table is (U)

2.3.3.2.2.1.1.5 (U) Cost/Limitations

(U//FOUO) Moving HAIPE back to end systems may not be as economical as fronting multiple end systems with a single HAIPE INE.

2.3.3.2.2.1.1.6 (U) Dependencies

(U//FOUO) Key management is needed to support commercial IPsec and HAIPE implementations. HAIPE supports Pre-Placed Keys (PPKs) as well as auto-generated Traffic Encryption Keys (TEKs). Auto-generation includes FIREFLY and Enhanced FIREFLY.

(U//FOUO) HAIPE also depends on remote security management via a Security Management Workstation (SMW).

2.3.3.2.2.1.1.7 (U) Alternatives

(U//FOUO) FNBDT application layer security can be used to provide end-to-end protection for secure voice traffic.

(U//FOUO) Sub-network layer security can be used to protect information as it crosses a sub-network. Sub-network layer security allows black side switches, but still requires all IP routers to be red. For example, the CANEWARE Front End had a mode where it encrypted the payload of X.25 packets. A more modern example is FASTLANE, which provides security of the payload of ATM cells.

(U//FOUO) It is also possible to tunnel red side sub-network, link and physical layers over a black IP network. For example, BLACKER provided the ability to map red side X.25 addresses to black side IP addresses creating a Red Virtual Network which spanned a black side internet. Additional examples include the NES and Sectera INEs, which provide the ability to map red side MAC addresses to black side IP addresses essentially bridging a black side internet.

(U//FOUO) Security is also possible over SONET using the KG-189 to provide security of the SONET payload.

2.3.3.2.2.1.1.8 (U) Complementary Techniques

(U//FOUO) Link and physical layer mechanisms provide additional security. TRANSEC mechanisms support LPI/LPD. HAIPE has some TFS mechanisms, but link security can be used to enhance Traffic Flow Security (TFS). Higher layer mechanisms (e.g., S/MIME v3 for secure e-mail and FNBDT for secure voice) can be used to provide true end-to-end security and confidentiality within a domain.

2.3.3.2.2.1.2 (U) Traffic Flow Security (TFS)

(U) EDITOR'S NOTE: MATERIAL ON TFS WILL BE INCLUDED IN A FUTURE RELEASE

2.3.3.2.2.1.3 (U) Virtual Private Networks (VPN)

(U//FOUO) A Virtual Private Network (VPN) generally connects two private networks over a publicly accessible network (e.g., the Internet). Most VPNs are IP implementations that can be handled by a company's existing Internet technology. A VPN can provide a secure connection between remote sites without additional expenses for leased lines, ISDN, or frame-relay and Asynchronous Transfer Mode (ATM) technologies.

2.3.3.2.2.1.3.1 (U) Technical Detail

(U//FOUO) VPNs provide authentication, integrity, and confidentiality security services across a network, usually a publicly accessible network. Most VPN products use IPsec to carry out these security features, but other protocols (e.g., SSL) are also used in some products. For non-IP networks, (e.g., Internetwork Packet Exchange [IPX] or AppleTalk) Layer 2 Tunneling Protocol (L2TP) is more suitable.

(U) IPsec VPNs are a network layer technology. This means they operate independent of the applications that may use them. Tunnel mode IPsec encapsulates the IP data packet, hiding the application protocol information. Once the IPsec tunnel is created, various connection types (e.g., web, email, VoIP, FTP) can flow through the tunnel, each destined for different destinations on the other side of the VPN gateway.

(U) SSL VPNs are a remote access solution because they do not require IT departments to upgrade and manage client software. All a user needs is a Web browser.

(U//FOUO) VPN products can be grouped into three categories:

- (U) Hardware-based systems
- (U) Firewall-based systems
- (U) Stand-alone application packages.

(U//FOUO) Hardware-based VPNs typically use encryption routers providing IP services, such as IPsec tunneling. This is a common deployment strategy in a corporate network infrastructure to securely connect remote networks. Another hardware implementation involves VPN gateways used as IPsec tunnel endpoints. The VPN gateways provide firewalls and routing, as well as authentication, encryption, and key management capabilities.

(U//FOUO) Firewall VPNs take advantage of a firewall's authentication and access control features adding a tunneling capability and encryption functionality.

(U//FOUO) Stand-alone application VPNs use software to perform the access control, authentication and encryption needed for the VPN. The software VPN solution is the least expensive but generally has the worst performance. Software VPNs are adaptive to technology changes because no hardware changes are involved. The software VPN is ideal for company employees working on travel or from home.

2.3.3.2.2.1.3.2 (U) Usage Considerations

(U//FOUO) When setting up a VPN, you must consider the following options:

- (U) Security protocols supported
- (U) Cryptographic algorithms supported
- (U) Key management system used
- (U) User authentication used
- (U) Server platforms that run the product
- (U) Client platforms supported
- (U) Accreditation or approval
- (U) Price and maintenance costs
- (U) Number of users or connections supported

2.3.3.2.2.1.3.2.1 (U) Implementation Issues

(U//FOUO) Commercial venders have VPN capabilities built into firewall, gateway, or router products. There are currently dozens of different COTS VPN products available today. These products range from supporting small business connections to supporting large organizations requiring thousands of connections. Many vendors have a family of VPN products that support the different ranges of user needs. Some products support modular upgrades and have integrated hardware VPN acceleration capabilities, delivering highly scalable, high-performance VPN services.

(U//FOUO) As the VPN products have advanced, their configuration and administration has become easier. Configuration and management tools have been created to make the establishment, administration and monitoring of VPN clients and networks easier to perform. Some products advertise a One-click VPN, where VPNs can be created with a single operation by using VPN communities. As new members are added to the community, they automatically inherit the appropriate properties and can immediately establish secure IPsec/IKE sessions with the rest of the VPN community.

(U//FOUO) IPsec is still the most popular protocol for performing VPN security, but SSL has been gaining support in the last few years. Many VPN vendors now support both protocols in either the same product or as a separate item. One advantage of SSL over IPsec is that SSL does not require special VPN client software on remote PCs, which reduces administrative costs.

(U) VPN Products

(U) There are many COTS VPN products. The following is a list of some of the VPN products available today:

- (U) Cryptek's DiamondTEK has been evaluated and validated in accordance with the provisions of the NIAP Common Criteria Evaluation and Validation Scheme and the Common Criteria Recognition Arrangement as a EAL4 product - http://www.cryptek.com/SecureNetworks/
- (U) Blue Ridge Networks' VPN CryptoServer is also on the Common Criteria validated products list (EAL2) - http://www.blueridgenetworks.com
- (U) Check Point's VPN-1 Pro product line is an integrated VPN and firewall gateway, which offers management capability, attack protection and traffic shaping technology. - http://www.checkpoint.com/products/index.html
- (U) Nortel Networks Contivity is a line of VPN switches and gateways with supporting configuration and management tools. - http://www.nortelnetworks.com/products/family/contivity.html
- (U) Cisco PIX Security Appliances support hardware and software VPN clients, as well as PPTP and L2TP clients. - http://www.cisco.com/en/US/products/hw/vpndevc/index.html
- (U) SafeNet's HighAssurance™ Gateway product lines provide IPsec VPN solutions for small to large customers. SSL VPN also supported. - http://www.cylink.com
- (U) Avaya Secure Gateway products have specialized support for voice-over-IP (VoIP) applications - http://www1.avaya.com/enterprise/vpn/sg203_sg208/
- (U) Symantic Gateway supports both IPsec and SSL based VPN products - http://enterprisesecurity.symantec.com/content/productlink.cfm?EID=0
- (U) SonicWall has firewall and gateway products that feature IPsec VPN security - http://www.sonicwall.com/products/vpnapp.html
- (U) ADTRAN's NetVanta 2000 Series is a family of VPN/firewall appliances - http://www.adtran.com
- (U) ArrayNetworks Array SP family of appliances offers SSL VPNs - http://www.arraynetworks.net/globalaccess.htm
- (U) Celestix's RAS3000 supports SSL VPN for Microsoft Exchange Server 2003 - http://www.celestix.com/products/ras/ras3000/sslvpnforexchange.htm
- (U) Lucent Technologies Access Point® supports routing, secure VPN, QoS, firewall security, and policy management - http://www.lucent.com/solutions/
- (U) Juniper Networks Netscreen SSL VPNs provide a broad range of SSL VPN appliances - http://www.juniper.net/products/ssl/
- (U) V-ONE produces both IPsec and SSL VPN products - http://www.v-one.com

2.3.3.2.2.1.3.2.2 (U) Advantages

(U//FOUO) VPNs provide economical and secure solutions for remote access users (mobile users and telecommuters), intranets (site-to-site connections within a company or organization), and extranets (organization to organization network connections to suppliers, customers, or partners).

2.3.3.2.2.1.3.2.3 (U) Risks/Threats/Attacks

(U//FOUO) VPN clients should not be able to access your private network and the Internet at the same time. Doing so can be a security risk if the VPN client can become a gateway between the Internet and the private network.

(U//FOUO) PPTP authentication dependence on Microsoft Challenge Handshake Authentication Protocol (MSCHAP) makes it vulnerable to attacks using a hacker tool called l0phtcrack.

(U//FOUO) Nearly all computer equipment is susceptible to Distributed Denial of Service (DDoS) attacks. The Corrent Corporation's S3500 Turbocard Firewall/VPN accelerator is one VPN product that can withstand a massive DDoS attack, while keeping valid network traffic flowing at a high rate. The new Corrent® S3500 Turbocard is able to sustain 50,000 TCP sessions per second and deliver 648 Megabits per second in throughput in the face of a concentrated attack.

2.3.3.2.2.1.3.3 (U) Maturity

(U//FOUO) VPNs are a mature technology in the commercial world and continue to evolve. Products continue to support additional protocols and algorithms and run on more and more different platforms.

(U//FOUO) Interoperability between different manufacturers has seen significant improvements over the last few years but interoperability issues still exist.

(U//FOUO) VPNs are Mature (TRLs 7 – 9). Interoperability between different manufacturers and platforms should continue to move forward.

2.3.3.2.2.1.3.4 (U) Standards

(U//FOUO) Table 2.3-10 identifies the standards applicable to VPN technology.

Table 2.3-10: (U//FOUO) Standards Applicable to VPN Technology

This table is (U)

Number	Title	Date
RFC-2401	Security Architecture for the Internet Protocol http://www.ietf.org/rfc/rfc2401.txt	November 1998
	http://www.ietf.org/internet-drafts/draft-ietf-ipsec-rfc2401bis-02.txt	April 2004
RFC-2402	IP Authentication Header http://www.ietf.org/rfc/rfc2402.txt	November 1998
	http://www.ietf.org/internet-drafts/draft-ietf-ipsec-rfc2402bis-07.txt	March 2004
RFC-2406	IP Encapsulating Security Payload (ESP) http://www.ietf.org/rfc/rfc2406.txt	November 1998
	http://www.ietf.org/internet-drafts/draft-ietf-ipsec-esp-v3-08.txt)	March 2004
	The SSL Protocol, Version 3.0 http://wp.netscape.com/eng/ssl3/ssl-toc.html	November 1996
RFC 3031	Multiprotocol Label Switching Architecture http://www.ietf.org/rfc/rfc3031.txt	January 2001
RFC 2661	Layer Two Tunneling Protocol (L2TP) http://www.ietf.org/rfc/rfc2661.txt	August 1999
RFC 2637	Point-to-Point Tunneling Protocol (PPTP) http://www.ietf.org/rfc/rfc2637.txt	July 1999
	VPN Protection Profile for Protecting Sensitive Information http://www.iatf.net/protection_profiles/file_serve.cfm?chapter=vpn_pp.pdf	February 2000

This table is (U)

2.3.3.2.2.1.3.5 (U) Cost/Limitations

(U//FOUO) Most VPN products have a maximum connection number. So before purchasing a VPN product you must determine the maximum number of VPN connections you expect to have.

2.3.3.2.2.1.3.6 (U) Alternatives

(U//FOUO) There are several alternatives to VPN security of information over an untrusted network.

- (U//FOUO) FNBDT application layer security can be used to provide end-to-end protection for secure voice traffic
- (U//FOUO) Sub-network layer security can be used to protect information as it crosses a sub-network. Sub-network layer security allows black side switches, but still requires all IP routers to be red. For example, the CANEWARE Front End had a mode where it encrypted the payload of X.25 packets. A more modern example is FASTLANE, which provides payload security for ATM cells

- (U//FOUO) Security is also possible over SONET using the KG-189 to provide security of the SONET payload.

2.3.3.2.2.1.3.7 (U) References

(U) http://www.cryptek.com/SecureNetworks/

(U) http://www.blueridgenetworks.com

(U) http://www.checkpoint.com/products/index.html

(U) http://www.nortelnetworks.com/products/family/contivity.html

(U) http://www.cisco.com/en/US/products/hw/vpndevc/index.html

(U) http://www.cylink.com

(U) http://www1.avaya.com/enterprise/vpn/sg203_sg208/

(U) http://enterprisesecurity.symantec.com/content/productlink.cfm?EID=0

(U) http://www.sonicwall.com/products/vpnapp.html

(U) http://www.adtran.com

(U) http://www.arraynetworks.net/globalaccess.htm

(U) http://www.celestix.com/products/ras/ras3000/sslvpnforexchange.htm

(U) http://www.lucent.com/solutions/

(U) http://www.juniper.net/products/ssl/

(U) http://www.v-one.com

(U) Security Architecture for the Internet Protocol - http://www.ietf.org/rfc/rfc2401.txt and http://www.ietf.org/internet-drafts/draft-ietf-ipsec-rfc2401bis-02.txt

(U) IP Authentication Header - http://www.ietf.org/rfc/rfc2402.txt and http://www.ietf.org/internet-drafts/draft-ietf-ipsec-rfc2402bis-07.txt

(U) IP Encapsulating Security Payload (ESP) - http://www.ietf.org/rfc/rfc2406.txt and http://www.ietf.org/internet-drafts/draft-ietf-ipsec-esp-v3-08.txt)

(U) The SSL Protocol, Version 3.0 - http://wp.netscape.com/eng/ssl3/ssl-toc.html

(U) Multiprotocol Label Switching Architecture - http://www.ietf.org/rfc/rfc3031.txt

(U) Layer Two Tunneling Protocol (L2TP) - http://www.ietf.org/rfc/rfc2661.txt

(U) Point-to-Point Tunneling Protocol (PPTP) - http://www.ietf.org/rfc/rfc2637.txt

(U) VPN Protection Profile for Protecting Sensitive information - http://www.iatf.net/protection_profiles/file_serve.cfm?chapter=vpn_pp.pdf

2.3.3.2.2.2 (U) Real-Time Data Technologies

2.3.3.2.2.2.1 (U) Secure VoIP Call Control

(U//FOUO) Secure VoIP Call Control addresses technologies used to protect the signaling plane or VoIP call establishment protocols. Secure VoIP technologies that focus on the voice packets are described in the previous secure VoIP section.

(U//FOUO) Note that Secure VoIP call control mechanisms may be considered part of network management and control technologies within the GIG. Further information on VoIP call controls from the view of network security may be addressed in the GIG network control technology section. This section concerns itself with the protection of user information that is potentially vulnerable when using VoIP call control. This section also complements the secure VoIP section that describes methods to secure user voice.

2.3.3.2.2.2.1.1 (U) Technical Detail

(U//FOUO) A brief introduction to VoIP call control is presented followed by a description of security mechanisms that can be used to secure call control.

(U) The most common call control protocols used in VoIP system today include:

- (U) Session Initiation Protocol (SIP)
- (U) H.323
- (U) Media Gateway Control Protocol (MGCP)
- (U) Gateway Control Protocol (GCP), formerly MEGACO, and H.248

(U) In the spirit of distributed call control rather than centralized, integrated call managers, the IETF has decomposed gateways into Media Gateway Controllers and Media Gateways. As such, MGCP and GCP are protocols that can be used when PSTN-VoIP Gateways (PSTN-VoIP GWs) and multi-media conference units are decomposed between control and media processing units. This document does not address MGCP and GCP security mechanisms. It is assumed that GW control and processing units are integrated or collocated within a single security domain such that security mechanisms do not need to be applied to MGCP or GCP. Commercial IPsec or TLS could be used to secure these protocols within a single security domain if needed.

(U//FOUO) VoIP networks also require a QoS architecture designed to support the voice service. As such, secure mechanisms for the GIG QoS architecture needs to be developed as a complementary technology for secured voice. GIG secure VoIP call control and secure QoS mechanisms may likely need to work with each other, possibly thorough a network interface. Secure QoS security mechanisms are beyond the scope of this section.

(U//FOUO) Priority of Service, PoS, is another important voice feature. This feature allows users to pre-empt other voice calls or be placed in a higher priority queue for call processing. The GIG secure VoIP call control will need to request or invoke priority of service and, therefore, is likely to interact with GIG PoS security mechanisms. PoS security mechanisms are beyond the scope of this section.

(U//FOUO) Also note that the GIG VoIP call control architecture will need to include a dialing plan that not only includes SIP and H.323-based user identities, but also users on non-GIG networks expected to conform to E.164 numbering plans. This implies directory service techniques such as Electronic Numbering (ENUM). As such, GIG directory services that provide VoIP calling plans will need to be secured. The VoIP call control security mechanisms will need to interact with the security mechanisms of these directory services. GIG directory service security technologies are beyond the scope of this section.

(U) Therefore, this section focuses on SIP and H.323 as addressed below.

(U) SIP

(U) SIP is a text based client/server protocol that can establish, modify, and terminate multimedia sessions (conferences) or Internet telephony calls. SIP can invite participants to unicast and multicast sessions. An initiator does not necessarily have to be a member of the session to which it is inviting, and new media streams and participants can be added to an existing multi-media session. It can establish, modify, and terminate multimedia sessions or calls, such as conferences, Internet telephony and similar applications. SIP enables VoIP gateways, client end points, Private Branch Exchanges (PBX), and other communications systems and devices to communicate with each other.

(U) SIP transparently supports name mapping and redirection services, allowing the implementation of Intelligent Network telephony subscriber services. These facilities also include mobility that allows the network to identify end users as they move.

(U) SIP supports five facets of establishing and terminating multimedia communications:

- (U) User location: determination of the end system to be used for communication
- (U) User capabilities: determination of the media and media parameters to be used
- (U) User availability: determination of the willingness of the called party to engage in communications
- (U) Call setup: ringing, establishment of call parameters at both called and calling party
- (U) Call handling: including transfer and termination of calls.

(U) SIP does not offer conference control services such as floor control or voting and does not prescribe how a conference is to be managed, but SIP can be used to introduce conference control protocols. SIP does not allocate multicast addresses and does not reserve network resources.

(U) SIP network elements are shown in Figure 2.3-20 and described below.

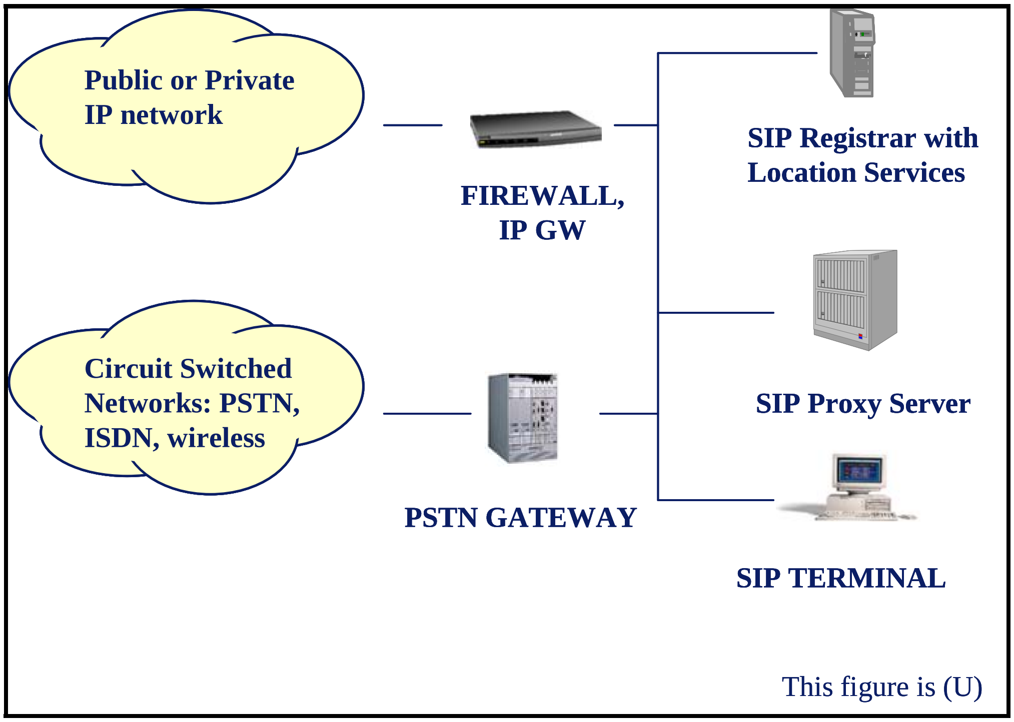


Figure 2.3-20: (U) SIP Architecture

(U) USER AGENT: The user agent, shown here as a client, accepts requests from a user and provides the appropriate SIP messages, or receives SIP messages and provides appropriate responses to the user. It is the SIP end point in the network. Examples for such a user agent might be an SIP-enabled PC or a SIP-enabled UMTS mobile device. Gateways can also act as SIP user agents, for example a VoIP SIP phone calling a POTS line will connect to the PSTN via a VoIP GW. The VoIP GW, then provides the SIP endpoint, or user agent, for the POTS line.

(U) REGISTRAR: The SIP Registrar is a server that receives registration requests from USER AGENTS in order to keep a current list of all SIP users and their location which are active within its domain/network. A registrar is typically collocated with a proxy or redirect server.

(U) LOCATION SERVICES: Location services find the location of a requested party in support of SIP based mobility. For example, when a SIP agent places a session or call request to another network user, the SIP location server will find the domain in which the second party was last registered. When found, the SIP request (SIP INVITE to the session) will be forwarded to the appropriate domain.

(U) PROXY SERVER: The proxy server is an intermediate device that receives SIP requests from a user agent and then forwards the requests on the client's behalf. The proxy server can stay in a signaling path for the duration of the session. Proxy servers can also provide functions such as authentication, authorization, network access control, routing, reliable request retransmission, and security. Equally important, the proxy server can be used by the network to execute a range of supplementary services. Soft switches, for example, may use the SIP proxy as a way to interface the call or session model to the user agent. In the VoIP world, call forwarding on busy can be implemented and invoked in the proxy server.

(U) RE-DIRECT SERVER: The re-direct server provides the client with information about the next hop or hops that a message should take, and then the client contacts the next hop server or user agent directly. Unlike the use of a proxy server, the re-direct server simply serves to direct on-going communications at session initiation, but does not stay in the signaling path for the duration of the data session.

(U) SIP Security Mechanisms

(U) The SIP RFCs describe a number of security mechanisms that can be used within a SIP system. Message authentication and encryption of SIP messages are supported. Since SIP uses proxy servers and registrars in support of service execution on behalf of the user, SIP assumes security associations between the SIP client and various SIP infrastructure elements, rather than secured end-to-end call control security between voice users. This, of course, means that all SIP servers need to be trusted network elements. As such, all SIP call control is assumed to be located within a single security domain. IPsec can be used to tunnel SIP call control from SIP clients to SIP servers across backhaul networks that may be outside the SIP security domain. Note that SIP is a text-based protocol, borrowing many elements from other text based protocols such as HTTP. As such, SIP security reuses HTTP and MIME security mechanisms as explained below. (Note that PGP is not longer recommended in the latest SIP RFC.)

(U) The following encryption scenarios are identified in the SIP RFCs:

- (U) A network can use lower layer security protocols, such as IPsec or TLS between the SIP UA and SIP server. Although many implementations transport SIP with UDP, SIP can also be transported with TCP so that TLS can be used
- (U) TLS can be used between servers
- (U) S/MIME techniques can be used to encrypt SIP bodies for end-to-end security of the SIP message payload, while leaving SIP headers in the clear for server support. S/MIME also provides for integrity and supports mutual authentication. Note that this method does offer protection of user network address or URI information. Furthermore, specific applications of SIP servers can offer the user a number of network enabled services. The set of services offered by these kinds of SIP servers may be restricted if the SIP message body is opaque to the SIP server

(U//FOUO) SIP includes basic password authentication mechanisms as well as digest based mechanisms. The SIP protocol includes messages that facilitate authentication. For example, SIP protocol specific authentication challenge messages Response 401 (Unauthorized) or Response 407 (Proxy Authentication Required) can be used in conjunction with a cryptographic mechanism. The Digest authentication mechanisms called out by the SIP RFCs are based upon HTTP authentication.

(U) SIP also defines option mechanisms to convey user privacy requirements. Finally, SIP extensions are identified for media and QoS authorization as a means of protecting against DoS attacks.

(U) A Brief Overview of H.323

(U) H.323 from the ITU is binary based (ASN.1) and provides for both signaling plane messaging as well as signaling to bearer control. Because it was initially designed to support video packets, H.323 has considerable overhead, which is a disadvantage for IP telephony applications.

(U) Note that H.323 is an umbrella specification, covering several protocols related to call setup and signaling. Chief among these are H.225, which defines the call signaling channel, H.245, or the call control channel, and RAS - registration, admission, status. Underlying these are the Real Time Transport Protocol (RTP) and/or the Real Time Control Protocol (RTCP), which define the basic requirements for transporting real time data over a packet network.

(U) The H.323 architecture and components are introduced in Figure 2.3-21 and summarized below.

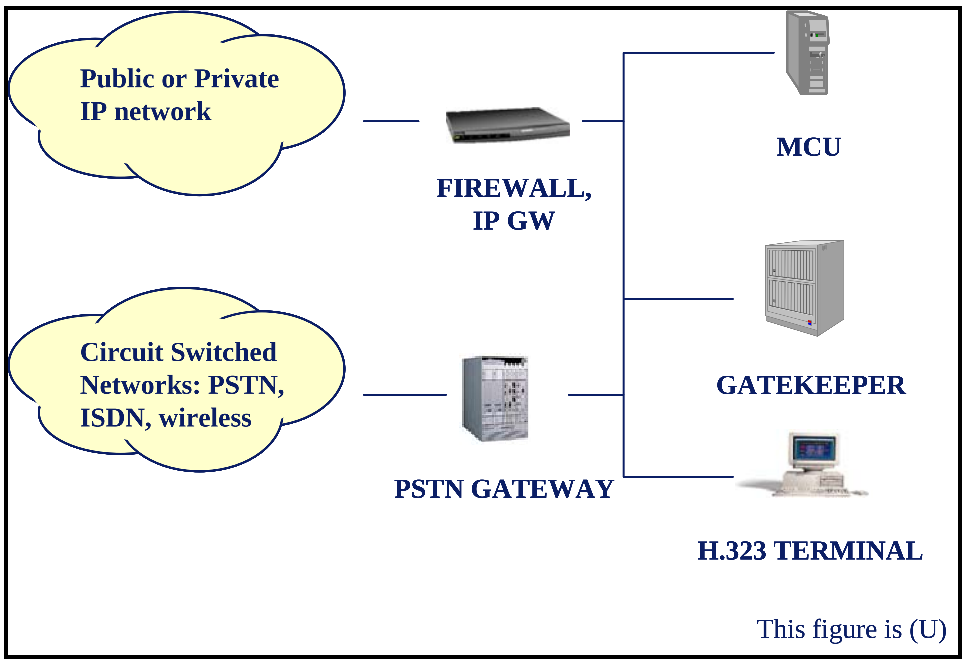


Figure 2.3-21: (U) H.323 Network Elements

(U) Gatekeeper:

- (U) Manages an H.323 zone or network (collection of H.323 devices)
- (U) Supports address translation, access and admissions control, bandwidth control, and allocation
- (U) Optional functionality includes call authorization, supplementary services, directory services, call management services

(U) Gateway

- (U) The Gateway provides interoperability between different networks by converting signaling and bearer between, as an example, IP and circuit-based networks
- (U) H.323 Terminal
- (U) The Terminal is the H.323 signaling endpoint/client on an IP network
- (U) Supports real-time, 2-way communications with another H.323 entity. Must support voice (audio codecs) and signaling (Q.931, H.245, RAS). Optionally supports video and data, e.g., PC phone or videophone, Ethernet phone

(U) Multipoint Control Unit (MCU)

- (U) The MCU supports conferences between 3 or more endpoints
- (U) The MCU must contain multi-point controller (MC) for signaling and may contain multi-point processor (MP) for media stream processing. The MCU can be stand-alone or integrated into gateway, gatekeeper or terminal

(U) H.323 Security Framework defined in H.235

(U) The H.235 standard defines a security framework to be used within H.323 systems. H.323 supports a menu of encryption, and authentication options are supported. Note that H.235 defines security mechanisms between H.323 clients and H.323 call control servers (Gatekeepers, MCUs, Gateways). End-to-end call control security between clients is not provided. Therefore, H.235 requires all H.323 infrastructure elements to be trusted servers. The H.245 signaling protocol used within H.323 systems includes methods to negotiate the security algorithms and keys used for a secure call control connection.

(U) H.235 supports DES, 3DES and AES encryption algorithms. H.235 allows for TLS or IPsec to be used amongst H.323 clients and H.323 call control servers.

(U) A variety of authentication options are identified, which include HMAC-SMA1-96. Public certificates as well as subscription-based authentication mechanisms can be used. Three options for subscription-based authentication are identified, specifically:

- (U) Password-based with symmetric encryption (shared secret)
- (U) Password-based with hashing
- (U) Certificate based subscriptions with digital signatures.

(U) Key management is incorporated into the H.323 family of signaling specifications. H.235 describes the use of H.245 signaling protocol messages for key management. The use of H.323 RAS protocol for key management has been identified in the specification, but is not completely defined. Third party key escrow schemes are described, and Diffe-Hellman can be use for key agreements.

(U) This menu of security options is organized within three security profiles as listed below:

- (U) The Basic security profile employs user passwords as part of the authentication approach
- (U) An optional Digital Signature profile utilizing certificates
- (U) A Hybrid profile that combines elements of Basic and Digital Signature

(U) H.235 also describes secure call control consideration in the presence of firewalls and Network Address Translation devices.

(U) Finally, H.235 also references H.510 and H.530, which together describe security mechanisms for mobility across H.323 systems. These specifications describe a generic security concept for mobility among domains. Hop-by-hop security with shared secrets is employed to protect call control.

2.3.3.2.2.2.1.2 (U) Usage Considerations

2.3.3.2.2.2.1.2.1 (U) Implementation Issues

(U) Call Control Environment

(U//FOUO) Since both SIP and H.323 security measures rely upon trusted call control network elements to complete call processing, the VoIP call control will need to reside within a single security domain. Furthermore, SIP and H.323 security mechanisms are not type 1 and do not lend themselves to existing type 1 solutions. Therefore, other security mechanisms, such as HAIPEs, are required to transport call control across security domain boundaries. For example, Secure SIP mechanisms at the session layer can be used amongst devices within the security domain. Clients roamed onto other security domains can use HAIPEs to tunnel back into the security domain to join VoIP calls using the SIP security. As such, the client would apply SIP security over HAIPEs security for the call control.

(U) Clear – Secure Transitions

(U//FOUO) Note that the ability to transition between clear and secure voice is an important security feature for voice services. As such, the GIG VoIP architecture needs to allow for a transition between clear and secure voice media types within a single call.

(U) Multiple call control systems and user mobility

(U//FOUO) Since SIP and H.323 need to reside in a single security domain, it is conceivable that a SIP client may need to operate with a number of call control managers, depending upon the security domain of other call participants. For example, a user may need to register on the GIG VoIP call control manager for communications with on-GIG users and another call control manager to communicate with a non-GIG or coalition user. This means that user mobility may need to be tracked in multiple call control planes, based upon security domain.

(U) Security technologies within the call control environments

(U) SIP security

(U//FOUO) The SIP security RFCs call out a menu of security options. Therefore, the GIG security architecture will need to standardize on a specific SIP security profile to ensure interoperability. This is especially important since call control needs to pass through a chain of trusted clients and trusted call control network elements (servers) in order to place a VoIP call. Furthermore, non-GIG networks may use other SIP security profiles, requiring GIG clients to support additional security mechanisms when communicating with non-GIG users.

(U//FOUO) SIP interaction with IPv6 firewalls is not clearly defined and needs to be studied.

(U//FOUO) Note that many VoIP networks place SIP on top of UDP. But TLS forces SIP to be placed on TCP, reducing the efficiency of the protocol in comparison with UDP-based approaches. IPsec may be more efficient alternative to TLS in this case.

(U) H.323 security

(U//FOUO) The implementation of H.323 security shares much of the same concerns as SIP security. H.323 offers a wide variety of security mechanisms within three profile types. The GIG security architecture will need to standardize on a specific set of security functions for interoperability. Although H.235 does address interaction with IPv4 firewalls, IPv6 firewall interaction requires further study.

(U) Unlike SIP, H.323 was originally designed for TCP, so the use of TLS amongst clients and servers fits well within the H.323 system.

2.3.3.2.2.2.1.2.2 (U) Advantages

(U) SIP security mechanisms fit well within a VoIP architecture and reuse many techniques from HTTP and S/MIME security.

(U) H.323 security is flexible and, like SIP, is intended to fit well within VoIP architectures.

2.3.3.2.2.2.1.2.3 (U) Risks/Threats/Attacks

(U//FOUO) The SIP RFCs declare SIP to be difficult to secure. SIP security requires trusted SIP infrastructure (proxies, registrars, etc) and hop-by-hop security mechanisms such as TLS to avoid security risks. Even so, SIP security features are not type 1 and would need to be extended to support type 1 techniques. This limits SIP to reside within a single security domain.

(U//FOUO) As with SIP, H.323 requires trusted call control infrastructure and hop-by-hop security to avoid security risks. Although H.235 describes a number of possible non-repudiation techniques, the overall topic of non-repudiation is listed for further study in the specification. Further security evolution is likely. As with SIP, H.323 is not defined to support type 1 security and would need to be extended to do so. This limits H.323 to reside within a single security domain.

2.3.3.2.2.2.1.3 (U) Maturity

(U//FOUO) Although some of the security techniques SIP leverages are well known and have deployed in commercial networks, the overall SIP security is immature and not widely used. Therefore, SIP Security as defined in the RFCs is Emerging with an estimated TRL of 5.

(U//FOUO) Similar to SIP, many of the H.323 security mechanism are used in deployed networks. But the overall H.235 security framework is not widely used in VoIP networks. As such, H.235 is also Emerging with an estimated TRL of 5.

2.3.3.2.2.2.1.4 (U) Standards

(U) Table 2.3-11 summarizes the standards relevant to secure VoIP call control.

Table 2.3-11: (U) Secure VoIP Call Control Standards

This table is (U)

Name	Description
H.235	Security and encryption for H-series multimedia terminals
H.245	Call Control Protocol for multimedia communication: Series H
H.323	Packet-based multimedia communications: Series H
H.510	Mobility for H.323 multimedia systems and services
H.530	Symmetric security procedures for H.323 mobility in H.510
RFC 3262	SIP: Session Initiation Protocol
RFC 3310	HTTP Digest Authentication Using Authentication and Key Agreement (AKA)
RFC 3313	Private SIP Extensions for Media Authorization
RFC 3323	A Privacy Mechanism for the SIP
RFC 3325	Private Extensions to the SIP for Asserted Identity within Trusted Networks
RFC 3329	Security Mechanism Agreement for the SIP
RFC 3435	Media Gateway Control Protocol
RFC 3525	Gateway Control Protocol
RFC 3761	The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM)
RFC 3762	Telephone Number Mapping (ENUM) Service Registration for H.323
RFC 3853	S/MIME Advanced Encryption Standard (AES) Requirement for the Session Initiation Protocol (SIP)

This table is (U)

2.3.3.2.2.2.1.5 (U) Cost/Limitations

(U//FOUO) SIP and H.323 security mechanisms require trusted call control network elements, restricting application of SIP and H.323 security to within a single security domain. Lower layer security, such as HAIPE, is required to tunnel SIP across security domain boundaries. These multiple layers of security add complexity to client call control processing and may help increase costs.

(U//FOUO) Note that there is no method defined to provide protection of circuit-based signaling. Therefore any protection offered SIP or H.323 would terminate at a VoIP-circuit signaling gateway.

2.3.3.2.2.2.1.6 (U) Dependencies

(U//FOUO) The level of trust of any proxy within the call control chain for either SIP or H.323 call control may impact RAdAC and policy enforcement.

(U//FOUO) The GIG key management architecture will need to take into account SIP and H.323 key management requirements. Although SIP does not specify key management systems, H.245 does include key manage messages.

(U//FOUO) The security mechanism of a VoIP architecture will likely need to interact with the security mechanisms applied to the GIG QoS and PoS architectures.

(U//FOUO) Also note that GIG directory services will need to accommodate GIG and off GIG ‘dialing plans’ to support VoIP call control. The protection of these directory services and VoIP call control security will need to interact and be coordinated within the GIG architecture.

2.3.3.2.2.2.1.7 (U)Alternatives

(U//FOUO) HAIPE could be applied not only to tunnel SIP across security domains, but could also be used to protect call control within a single domain. A HAIPE tunnel could be applied between clients and servers and between servers.

2.3.3.2.2.2.1.8 (U) Complementary Techniques

(U//FOUO) Protection of QoS, protection of PoS, protection of directory services, and the use of HAIPEs to span security domains are complementary technologies to secure VoIP call control.

(U//FOUO) A secure key management system that accommodates VoIP call control security mechanisms is also a complementary technology.

2.3.3.2.3 (U) Link & Physical Layer Technologies

2.3.3.2.3.1 (U) Anti-Jam

(U) EDITOR'S NOTE: MATERIAL ON ANTI-JAM WILL BE ADDED IN A FUTURE RELEASE.

2.3.3.2.3.2 (U) Link Encryption

(U) EDITOR'S NOTE: MATERIAL ON LINK ENCRYPTION WILL BE ADDED IN A FUTURE RELEASE. IT WILL ADDRESS IMPLICATIONS OF EXPANDING LINK ENCRYPTIONS CAPABILITIES TO MEDIUM AND LOW ASSURANCE LINKS.

2.3.3.2.3.3 (U) TRANSEC

(U) EDITOR'S NOTE: MATERIAL ON TRANSEC WILL BE ADDED IN A FUTURE RELEASE.

2.3.3.3 (U) Trusted Platforms

2.3.3.3.1 (U) Technical Detail

2.3.3.3.1.1 (U) Definition

(U) A trusted platform is a GIG component that is relied upon to enforce its security policy. That is, it has been assigned a set of security rules to enforce (its policy) and is relied upon to enforce those rules. No other GIG component will prevent a violation of the security policy if the trusted platform is subverted, successfully attacked, or otherwise fails to act appropriately.

(U) By contrast, an untrusted platform is not relied upon to enforce any specific policy. It is prevented from harming the GIG or its users by other trusted platforms.

(U) The security policy enforced by a given trusted platform can vary. In the 1980s, a trusted platform was considered to be one that could enforce a multilevel security policy. (See [TCSEC] for technical details.) That is, one could have information at different classification levels (e.g., SECRET information and TOP SECRET information) on the system at the same time, and possibly have users with different clearances accessing the system at the same time. And, one could have some appropriate level of confidence that a confidentiality policy would be correctly enforced. Examples: that TOP SECRET information would not be disclosed directly or indirectly to a user with only a SECRET clearance or that TOP SECRET and SECRET information would not be co-mingled in a file labeled SECRET. There might also be an integrity policy of some sort enforced, e.g., it might be prohibited for a SECRET user to modify or delete a TOP SECRET file.

(U//FOUO) With the GIG's Task Post Process Use (TPPU) model and different operational/networking scenarios, the definition of a trusted platform has been expanded from this previous meaning. It now has the more generic meaning given above; that is, a trusted platform enforces whatever security policy it has been assigned. It does not have to be a traditional multilevel security policy. Some trusted platforms enforce MILS policies. These policies allow the platform to be used for different levels of security at different times—while restricting use to one level at any given time. For example, a device could be used to connect to an unclassified network and process unclassified information and then later be used to connect to a classified command and control network and process SECRET information.

(U//FOUO) The concept of MILS is similar to the traditional periods processing operations of processing one security level of information, wiping the system of any information (e.g., by removing disks, tapes, etc.), and then reloading it to process another level of information. However, it is not acceptable to have to physically change a GIG component (e.g., to remove a SECRET hard drive and replace it with an UNCLASSIFIED hard drive) given the need for network connectivity and communications. Thus, a MILS system must enforce a security policy that separates information and grants access appropriately, while not requiring significant reconfiguration.

(U) Trusted platforms have two types of mechanisms: functions and assurances. Functions are the things that the platform actually does to enforce its security policy. Typical security functions in a trusted platform include identification and authentication of users, access controls, and auditing of security relevant events.

(U) Assurance mechanisms are things used during the development and operation of the trusted platform to gain confidence that it actually will work correctly in its intended environment, and that it will not have hidden, undocumented, or unintended features that will allow the security functions to be subverted. Assurance mechanisms that can be used for trusted platforms include things like mapping levels of specification (to determine consistency in the development of the platform), adherence to software development standards and practices, and testing.

(U) The earliest work in trusted platforms was carried out from the late 1960's to the early 1980's. It led to the DoD Trusted System Evaluation Criteria (TCSEC), which was the DoD standard from 1985 until the late 1990s. Work from other organizations (such as NIST) and other countries (such as Canada, which published its own Canadian Trusted Computing Platform Evaluation Criteria, and the United Kingdom, Germany, France and the Netherlands, which jointly published the Information Technology Security Evaluation Criteria) led to the development during the 1990s of the harmonized Common Criteria (encapsulated in ISO Standard 15408, volumes 1-3). The Common Criteria (CC) are recognized by at least 20 countries, including the U.S., and evaluation of a product against the Common Criteria is required now for use in most DoD programs.

(U) The CC intend for an organization (for example, an industry standards group or an organization interested in acquiring trusted platforms) to publish a Protection Profile. A Protection Profile is a set of security functions, drawn from ISO 15408 volume 2—combined with a set of required assurance mechanisms, drawn from ISO 15408 volume 3. It represents the set of requirements that a category of IT products must meet to be useful and secure. For example, there is a Protection Profile covering Multilevel Operating Systems in Environments Requiring Medium Robustness. This would be for any operating system that is to be used in multilevel secure operations, in environments where threats are non-trivial but not severe, must meet the requirements contained in that protection profile. Customers attempting to deploy systems in those environments should not use systems that have not been successfully evaluated against that Protection Profile.

((U) The CC evaluation scheme does allow for the evaluation of products even when there is no established Protection Profile. The vendor publishes a Security Target, which is a description of the security properties and capabilities of the product, and the product is evaluated against that Security Target. Potential customers can then review the Security Target to determine if the product is useful in their solutions.

(U) The functional and assurance mechanisms covered in ISO 15408 are largely independent. However, some dependencies are known to exist. Some of these exist because it is not possible to implement one function without another one (e.g., mandatory access controls cannot be enforced unless data objects are appropriately tagged/labeled). Others exist because the approximately 30 years of experience in this area indicates that they provide equivalent and compatible security. In ISO 15408-2, dependencies among the functional requirements are identified, and Protection Profiles that include a given requirement must also include all requirements on which the initial one depends.

(U) For the convenience of users, the assurance mechanisms in ISO 15408-3 have been grouped in a set of seven levels, referred to as Evaluated Assurance Level (EAL)—1 (the lowest) through EAL-7 (the highest). The intent is that each EAL-value describes a system that is usable in a specific type of environment. EAL-1 products tend to be appropriate in environments in which there is little to no threat; EAL-7 products are designed to stand up to the most rigorous threat environments known.

2.3.3.3.1.2 (U) Components

(U//FOUO) A trusted platform consists of hardware, software, and the guidance and procedures that go into using it. A trusted platform may be a COTS product (as most GIG components are expected to be) or it may be custom-designed device.

(U//FOUO) Security policy enforcement can be apportioned among the components of a trusted platform in any way the developer wishes. Typically, in a commercial trusted operating system, little to no enforcement is assigned to the hardware; all of the explicit enforcement functions are put into software. The hardware is merely relied upon to operate correctly; i.e., to operate in accordance with its specifications without having any ways in which the software policy enforcement can be avoided or prevented. For other solutions—including most Government-provided Information Assurance assets—the hardware is explicitly assigned security policy enforcement responsibilities, such as cryptography, tamper resistance, etc.

(U//FOUO) Trusted platforms must also include guidance/procedures for their assumed environments. For example, most commercial products do not offer strong tamper resistance. They are assumed to operate in environments in which modification or replacement of the hardware is prevented by physical and procedural security means. Operating a trusted platform outside of its assumed environment tends to negate the basis for trust in the system. Thus, guidance/procedures must be clear.

2.3.3.3.1.3 (U) Minimal Requirements

(U//FOUO) The requirements for specific trusted platforms to be used in the GIG will vary according the specific functions, roles and security policies of those platforms. However, there is a set of functions that must be supported in all cases for a platform to be considered trusted. This set includes:

- (U//FOUO) Identification and authentication of users, subjects, and objects. Different users of the system must be identified and authenticated. Other parts of the system—e.g., processes running as subjects; information objects contained within it—must be identified and, if appropriate, authenticated. It is important that the identification and authentication mechanisms cannot be subverted or bypassed by attackers. The strength of authentication mechanisms used for a specific platform will vary according to its uses For example, for some platforms, user ids and robust passwords will be sufficient; for others, biometrics or hardware tokens will be required. The strength of authentication mechanism required will generally be specified in the Protection Profile. If there is no Protection Profile covering this platform, the system engineering or requirements group will have to determine an appropriate level

- (U//FOUO) An ability to securely initiate (i.e., boot) the trusted platform. It must be possible to boot the system into a known secure state. This includes mechanisms that will verify the integrity of the system and its components during the boot process to detect modification/tampering. For example, the trusted platform may need to verify serial numbers or private keys assigned to hardware modules to ensure that they have not been removed or replaced (although strategies must exist to replace defective modules with new replacement or upgraded modules). Similarly, it may be appropriate to digitally sign boot code such as Basic Input-Output System (BIOS) code to ensure that it has not been modified. When using modification-detection routines as part of the secure initialization process, it is necessary to ensure that the detection routines themselves cannot easily be defeated. As noted, though, it is still also necessary to allow for required upgrades as well as the replacement of failed modules

- (U//FOUO) Partitioning. The trusted platform must have the ability to support different processes with different privileges. Those processes and the resources they access must be physically or logically partitioned, so that one process cannot interfere with or learn information about another process in violation of the security policy. Partitioning of the system supports MLS and/or MILS operation. It can be implemented using a virtual machine architecture, in which processes operating at one level are given access to virtual resources rather than the platform's physical resources (such as disk space, network interfaces, etc.). Partitioning usually requires that the platform have the ability to save process state, and to sanitize internal memory. It also requires that communications among processes operating at different security levels be controlled—whether simultaneously or at different times, since covert channels to leak information often exist in the ways processes interact

- (U//FOUO) Access control. The trusted platform must have the ability to support an access control policy. This policy determines what subjects may access what objects, in what contexts, and in what modes. In traditional MLS operation, this policy is label-based and follows the lattice model of security originally described by Bell and LaPadula. In MILS operation, typically all subjects can access all objects in the virtual machine that represents a single security level, but the virtual machine manager tightly controls accesses that cross or go beyond a virtual machine. In other operations, the policy can be based on integrity models, non-interference models, or other models. In addition, discretionary access control policies, in which object owners determine access rights, can also be enforced

- (U//FOUO) Auditing. The trusted platform must have the ability to track security relevant events that occur on the platform. These events will depend on the specific platform, environment, applications, and security policy to be enforced. See Section 2.7 of the Technology Roadmap for a description of Audit Management and Section 2.6 of the Technology Roadmap for a description of Computer Network Defense, which is closely related to audit and which all trusted platforms must support

2.3.3.3.1.4 (U) Implementing Trusted Platforms

(U//FOUO) There are a number of techniques that can be used to implement the functions of trusted platforms and to provide the assurance levels necessary to achieve the confidence that a trusted platform cannot be subverted. These techniques run from stronger policy enforcement mechanisms to implementation and testing techniques that increase confidence that bugs have been found. We will address these techniques in this section.

2.3.3.3.1.4.1 (U) Functions

(U//FOUO) Trusted platforms must identify and authenticate all users and subjects that access the system before allowing them to take any other security-relevant actions. The identification and authentication mechanisms chosen must be of appropriate strength—given the security requirements for the platform and the security mechanisms and assurances implemented for the rest of the system. Identification and authentication is discussed in detail in Section 2.1 of this document.

(U//FOUO) Trusted platforms must generally implement some form of access control. This could include one or more of: Rule-based or mandatory access control; discretionary access control; role-based access control, or other forms. In the future, it may be Risk-Adaptable Access Control (RAdAC), discussed in Section 2.2.

(U) Rule-based or mandatory access control is based on labels or metadata tags associated with subjects and objects. It is non-discretionary, in that access to an object can only be granted to a subject if the tags associated with the subject and object match according to the established rules. Data owners (originators) cannot change this. Typically, these access controls are used to implement classification-based access controls, e.g., to ensure that TOP SECRET data objects are only accessed by those with TOP SECRET clearances.

(U//FOUO) Discretionary access controls allow data object owners to make decisions about whether access is granted to a subject. Discretionary access controls provide a flexible mechanism, but they are vulnerable to attacks such as Trojan horses, in which a program acting as the data owner grant access without the owner's knowledge or approval.

(U//FOUO) Access controls are discussed in detail in Section 2.2 of the GIG IA Capability/Technology Roadmap.

2.3.3.3.1.4.2 (U) Assurance

(U//FOUO) More difficult than implementing functional mechanisms is achieving some level of assurance, that is, some level of confidence that the trusted platform will actually enforce its security policy and not be vulnerable to attack. A number of mechanisms can be used to accomplish assurance, some of which are more effective than others.

(U) In the Common Criteria (ISO 15408-3), assurance mechanisms are divided into seven classes:

- (U) Class ACM: Configuration management
- (U) Class ADO: Delivery and operation
- (U) Class ADV: Development
- (U) Class AGD: Guidance documents
- (U) Class ALC: Life cycle support
- (U) Class ATE: Tests
- (U) Class AVA: Vulnerability assessment

(U) Each of these classes has a number of mechanisms within it. Configuration management includes configuration automation, scope, and management. Delivery and operation includes requirements for secure delivery, installation, day-to-day operation, recovery, and platform life cycle support. Platform developers are required to provide guidance documents for users and operators/administrators that describe the intended environment and how to use, configure, and manage a trusted platform to meet its goals.

(U/FOUO) Life cycle support mechanisms include requirements for development environment security and for flaw remediation. Security of the development environment deals with the likelihood of malicious code or hardware being inserted into a trusted platform during its design and implementation. Flaw remediation deals with how a developer addresses security flaws once they are known. This includes reporting the flaw to customers; developing fixes for it; testing those fixes to ensure that they do fix the problem but do not cause additional security issues themselves; and distributing the fixes to customers.

(U) Testing includes both functional testing (e.g., making sure that a trusted platform meets all of its requirements) and independent penetration testing in which a Red Team attempts to attack a platform in an operational-type environment to look for security flaws that the development team did not contemplate. This independent testing is expanded in a vulnerability assessment, in which a developer and/or an independent Red Team analyze a trusted platform and attempt to identify all remaining flaws and vulnerabilities, so that informed choices can be made about whether to accept the vulnerabilities or to modify the system or environment to mask them.

(U/FOUO) That leaves the largest and most complex class of assurance mechanisms, which are the development mechanisms. These are the mechanisms that address how the system is to be designed and implemented. They include:

- (U) The functional specification of the trusted platform and its interfaces. This can be done in an informal style (e.g., written in natural language), a formal, mathematical way, or some combination

- (U) The high-level design of the trusted platform. This describes the platform in terms of major subsystems

- (U) The completeness of the implementation representation (that is, how accurately the completed trusted platform is represented by its documentation)

- (U//FOUO) The structure and correctness of the internals of the trusted platform. This includes requirements for structure and modularity to minimize the number and size of the components that must actually be trusted and allow for full analysis of them. There are also requirements for reducing or eliminating circular dependencies, and for minimizing non-security-critical code and hardware in security-critical modules. The goal is to make the trusted parts of the trusted component as small and simple as possible. This leads to components that will be less likely to have buffer overflows, incomplete implementations, etc.

- (U) The low-level design, which describes the trusted platform in terms of its component modules, their interfaces and dependencies. At this level, the design of the trusted platform is much more complete and much more representative of what will actually be fielded than is the high-level design described above

- (U) A demonstration of correspondence between the different levels of documentation (e.g., showing that the high-level design and low-level design are consistent with one another, and no gaps or major new areas exist in either document)

- (U) Security policy modeling. The purpose of this mechanism is to develop a model of the security policy to be enforced by the trusted platform and then to demonstrate that that model is actually enforced by the platform specification

(U//FOUO) Together, these mechanisms can be used to show that a trusted platform has been built with an acceptable level of confidence that it does not contain security vulnerabilities such as buffer overflows, incomplete or ineffective implementations, or undocumented features that can be used to defeat security.

2.3.3.3.2 (U) Usage Considerations

(U//FOUO) Trusted platforms have been around for more than 20 years. However, they have enjoyed essentially no success in the general computing field and very little success in special-purpose processing. Largely, the reasons for this have been the significant cost of these platforms and their user-unfriendliness.

(U//FOUO) It costs a tremendous amount of money to develop a strongly-secure trusted platform—typically many millions of dollars. And there is not now nor has there ever been a significant market for them. The market has typically been expressed in terms of thousands of units, at best. Thus, amortizing development costs has resulted in the charge to customers for these devices being many thousands of dollars per copy. Faced with this, most potential customers have chosen to try to find alternative solutions, and new customers have stayed away entirely.

(U//FOUO) Also, because of security restrictions and other design choices, trusted platforms generally present markedly different interfaces to users than do more general purpose systems. Users quickly become frustrated at slow interfaces, limited networking, and being unable to do what they are used to do on their home or office computers. Even though U.S. Government policy required the use of a low-level trusted platform for all computers purchased after 1992, they have never caught on, and user unhappiness is a major reason why.

(U//FOUO) With advances in research results and the move away from pure MLS to MILS and other simpler solutions, there is a better chance for trusted platforms now than there has been in the past. However, it will be a challenge for some time.

2.3.3.3.2.1 (U) Implementation Issues

(U//FOUO) The biggest implementation issues with trusted platforms are the cost to build and evaluate them, and the difficulty in providing an acceptable user interface. When used in many environments, trusted platforms prevent users from doing things in the way they have always done them (often for sound security reasons), and thus users will look for ways to circumvent the trusted platform or will choose other platforms entirely.

2.3.3.3.2.2 (U) Advantages

(U) The advantage of a trusted platform is that it allows a single device to be used to handle a variety of different processing needs across security domains. As the GIG causes security domains to be brought together into COIs, it will be important for some components to have a high level of trust. With a trusted component, a user can connect to unclassified sites and SECRET sites and TOP SECRET sites with the same device. This will lead to better information sharing and a clearer picture of the situation. With MLS capability, this information can then be shared across users having the same device. With a MILS solution, this sharing will be more limited, but it will still be a substantial improvement over what exists today.

2.3.3.3.2.3 (U) Risks/Threats/Attacks

(U//FOUO) No trusted platform is perfect, because we as a community do not have the technology to implement systems without bugs. All trusted platforms are subject to some security attacks. Some will exploit bugs in the design or implementation; others will go around the security policy (i.e., exploit system features that the trusted component was explicitly designed not to protect).

(U//FOUO) The biggest risk in the GIG is that a trusted platform will be relied upon too much. GIG security will always require defense-in-depth. Trusted platforms will have their roles, and they can provide great advantages. But they should never be used in environments outside those described in their documentation, and they should not be relied upon to provide perfect protection against attacks.

2.3.3.3.3 (U) Maturity Level

(U//FOUO) As noted above, trusted platforms have been around for more than 20 years. For some categories of products (e.g., firewalls, gateways, special purpose IA components), they are mature technology and can be used in the 2006-2008 GIG. For other categories of products (e.g., devices to connect to multiple levels of wireless networks; general purpose desktop computers), significant research is needed in the areas of software engineering, high-assurance computing, network security, and system evaluation. In addition, much work is needed for all types of platforms in the areas of system performance, user friendliness, and cost-effective security.

(U//FOUO) For those categories of products for which the technology is well adapted (e.g., firewalls and gateways), trusted platforms are Mature (TRLs 7 - 9). There are products which can be purchased and used today. In fact there are products that are being used within the DoD today.

(U//FOUO) For other types of products, significant research is needed. The technology readiness group is near the boundary of Emerging (TRLs 4- 6) and Early (TRLs 1 - 3).

2.3.3.3.4 (U) Standards

(U) Validated non-U.S. Government Protection Profiles on trusted platforms (per http://niap.nist.gov/cc-scheme/pp/index.html)

- (U) Trusted Computing Group (TCG) Personal Computer (PC) Specific Trusted Building Block (TBB) Protection Profile and TCG PC Specific TBB with Maintenance Protection Profile
- (U) Trusted Computing Platform Alliance Trusted Platform Module PP

(U) The primary standard for Trusted Platforms is the Common Criteria, ISO 15408, volumes 1-3. These documents are used as the basis for evaluation in the U.S. and approximately 20 other countries.

(U//FOUO) For other, non-COTS devices, there are specific standards internal to NSA that are applied to high-assurance devices.

2.3.3.3.5 (U) Cost/Limitations

(U//FOUO) As noted above, trusted platforms tend to be very costly to develop, and costly to use. Development costs are attributed to the fact that it (at least initially) costs a lot of money to build security into a product. Typical commercial best practices cannot be used, so the development system has to be changed. In addition, evaluation of a product costs money and time. Given that the market for these trusted platforms has traditionally been very small, the development costs have to be amortized over this relatively small base, and thus the cost to users tends to be high.

(U//FOUO) There is a perceived cost to the users in running a trusted platform in the GIG environment, as well. This cost is due to the fact that for security reasons the trusted platform often does not allow the users to do things the same way they've always done them.

2.3.3.3.6 (U) Dependencies

(U) As noted above, trusted platforms are dependent on a number of the other enablers: Identification and Authentication; Policy-Based Access Control; Network Defense and Situational Awareness; and Management of IA Mechanisms and Assets.

2.3.3.3.7 (U) Alternatives

(U//FOUO) The alternative to using trusted platforms is to restrict the GIG to having each device be used for a single classification level or domain of information and users. However, this defeats the GIG's vision of information sharing; it prevents RAdAC from being implemented; it drives up costs; and in general it will prevent the GIG from meeting its mission.

(U//FOUO) Within the field of trusted platforms, there are a variety of possible alternatives—traditional Multi-level security; Multiple Independent Levels of Security; various other security policies to be supported. It is likely that each of these alternatives will be useful for some set of scenarios, and thus that there will be a wide variety of trusted platforms deployed in the GIG in the future.

2.3.3.3.8 (U) Complementary Techniques

(U//FOUO) Trusted platforms can be deployed along with cross-domain solutions such as a firewall and gateways to provide cost-effective solution for the GIG. Trusted platforms can also be used with other IA components to strengthen security, e.g., a trusted platform along with a QoP-capable router provides better resource allocation for the GIG that resists attacks better than routers alone.

2.3.3.3.9 (U) References

(U) ISO 15408-1: Information technology — Security techniques — Evaluation criteria for IT security — Part 1: Introduction and general model, 1999.

(U) ISO 15408-2: Information technology -- Security techniques -- Evaluation criteria for IT security -- Part 2: Security functional requirements, 1999.

(U) ISO 15408-3: Information technology -- Security techniques -- Evaluation criteria for IT security -- Part 3: Security assurance requirements, 1999.

(U) Johnson, R. and D. Wagner, Finding User/Kernel Pointer Bugs with Type Inference, Proc. 13th USENIX Security Symposium, pp. 119-133, San Diego, CA, August 2004.

(U) “Static Disassembly of Obfuscated Binaries,” by Kruegel, C., W. Robertson, F. Valeur and G. Vigna, Proc. 13th USENIX Security Symposium, pp. 255 – 270, San Diego, CA, August 2004.

(U) “Protection Profile for Multilevel Operating System in Environments Requiring Medium Robustness,” Version 1.22, NIAP, May 23, 2001. Available at http://niap.nist.gov/cc-scheme/pp/PP_MLOSPP-MR_V1.22.html

(U) “Protection Profile for Single-level Operating Systems in Environments Requiring Medium Robustness,” Version 1.22, NIAP, May 23, 2001. Available at http://niap.nist.gov/cc-scheme/pp/PP_SLOSPP-MR_V1.22.html

(U) “Controlled Access Protection Profile,” Version 1.d, NIAP, October 8, 1999. Available at http://niap.nist.gov/cc-scheme/pp/PP_CAPP_V1.d.html

(U) “Labeled Security Protection Profile,” Version 1.d, NIAP, October 8, 1999. Available at http://niap.nist.gov/cc-scheme/pp/PP_LSPP_V1.b.html

(U) “Copilot – a Coprocessor-based Kernel Runtime Integrity Monitor,” Petroni, N., T. Fraser, J. Molina, and W. Arbaugh, Proc. 13th USENIX Security Symposium, pp. 179 – 193, San Diego, CA, August 2004.

(U) “Design and Implementation of a TCG-based Integrity Measurement Architecture,´ Sailer, R., X. Zhang, T. Jaeger, and L. van Doorn, in Proc. 13th USENIX Security Symposium, pp. 223 – 238, San Diego, CA, August 2004.

(U) TCSEC –Department of Defense Trusted Computer System Evaluation Criteria, DoD Standard 5200.28-STD (obsolete), December 1985. Available at http://www.radium.ncsc.mil/tpep/library/rainbow/5200.28-STD.html

2.3.3.4 (U) Trusted Applications

2.3.3.4.1 (U) Technical Detail

2.3.3.4.1.1 (U) Definition

(U) For the purposes of this Technology Roadmap, a trusted application is an application that is relied upon while performing its functions to enforce a specific security policy. The security policy may be as simple as not being malicious or it may involve simultaneously processing and separating information from several domains (e.g., at multiple classification levels).

(U) An application is simply a program with a specific task or use. That is, a database management system may be an application, or a web server, or a browser, or an e-mail program. An operating system, or other generic program without a specific task to perform, is not an application. (Trusted operating systems are addressed in section 2.3.3.2 of the GIG IA Capability/Technology Roadmap.)

(U) At a minimum, a trusted application is not and cannot be subverted by malicious logic, and it does not harm other components of the GIG—including the platform(s) on which it is hosted. Depending on its specific security policy, a trusted application may also have other responsibilities, such as the separation of classified information.

(U) There are different definitions of trusted applications available in the literature. Some of them vary only in syntax, others in semantics. For example, some definitions assume that a trusted application is one that simultaneously processes information at multiple level of security; i.e., the trust is conferred because of the way the application was designed and operates. Other definitions assume that a trusted application is one that a user has accepted and agrees to let run on a computer without restrictions; i.e., the trust is conferred because the user has accepted the application based on whatever evidence exists.

(U) Yet another definition involves the ability of a trusted application to do harm. That is, a trusted application is one that, by its design and implementation, could subvert the security of the GIG if it unintentionally or maliciously attempts to violate the security policy of its host platform. That is, the application is trusted because it has to be; it isn't confined by a platform (e.g., an operating system) in such a way that the damage it can cause is constrained. An untrusted application, by contrast, is one that is constrained by a trusted platform (Section 2.3.3.2 of this document) so that it cannot directly impact the security of the GIG by either malicious use by an attacker, or simply by error in its own implementation and operation.[8]

(U) The resolution of these conflicting ideas is to use the definition cited above. That is, a trusted application is one that is assigned a security policy and is relied upon to enforce that security policy.

[8] (U//FOUO) Note that, where availability is concerned, this means that a trusted application cannot prevent other applications or the system from meeting its security requirements. Any application can fail to provide availability for itself, simply by failing to work.

2.3.3.4.1.2 (U) Security Policies

(U) As noted above, a trusted application may have any of a variety of security policies. In the most extreme cases, a trusted application such as a trusted database management system (see [TDI] for more details) may have a significant security policy such as maintaining the separation of information at different classification levels, while at the same time allowing accesses by users with different clearance levels. At the other end of the spectrum, a trusted application may have a security policy of doing no harm. That is, the application is trusted to execute and perform its task, without carrying or being vulnerable to viruses, worms, or other malicious logic; and without being able to cause denial of service attacks on its host component or on other GIG components.[9]

2.3.3.4.1.3 (U) Host Platform

(U//FOUO) A trusted application runs on top of a host platform. This host platform consists of hardware and software (e.g., an operating system) that provides support for the application. Enforcement of the application's security policy depends directly on the host platform. The host platform provides basic facilities (e.g., storage, processing, access to printers and networks) that the application requires to enforce its security policy. The host platform can also be used as a vector for attacking a trusted application; e.g., an attacker can modify the processor to ensure that encryption routines are not called when requested, or are called with pre-determined keys and initialization vectors to prevent the proper encryption of data.

(U//FOUO) Because of this, there must be an appropriate match between the security policy assigned to the trusted application and the capabilities of the host platform on which the application executes. For example, it is generally NOT acceptable to run an application providing multilevel security on a host platform that does not support strong process separation and access control—it is too easy for the application to be subverted.

2.3.3.4.1.4 (U) Requirements for Trusted Applications

(U) The specific requirements that are to be imposed on any trusted application depend on both what the application is supposed to accomplish and what its security policy is. Regardless of those factors, though, a set of minimum requirements can be established for all trusted applications. These requirements provide a basis for establishing some level of confidence that the application will enforce its security policy.

(U) These requirements are levied upon the execution environment in which the application runs (e.g., the operating system and hardware on which a program runs), as well as on the application itself. Peinado, et al. [PEINADO] list the following properties that must be possessed by the application and its execution environment.

- (U) No interference: The execution environment must provide a program that executes in it with the same underlying machine interface every time the program executes. The program must be isolated from external interference. A necessary condition is that a

[6] (U//FOUO) Note that a trusted application is not "trusted" to work correctly. Program correctness – working correctly without error, under all input conditions – is not a function of Information Assurance as addressed here. A trusted application may still hang, or fail to complete; it may still calculate an incorrect answer or provide incorrect outputs for specific data.

deterministic sequential program that does not access devices or persistent state should always reach the same result, irrespective of other programs that might have executed earlier or at the same time on the same machine

- (U) No observation: The computations and data of a program should not be observable by other entities, except for data the program chooses to reveal (e.g., through IPC)
- (U) Trusted paths: A program should be able to receive data from a local input device (e.g., keyboard, mouse), such that only the program and the user of the input device share the data. Data integrity must be assured. A similar requirement applies to local output devices, such as video.
- (U) Persistent storage: A program should be able to store data (e.g., cryptographic keys) persistently, so that the integrity and the confidentiality of the data are ensured
- (U) Communication: A program should be able to exchange data with another program, such that the integrity and the confidentiality of the data are ensured
- (U) Local authentication: A local user should be able to determine the identity of a program
- (U) Remote authentication: A program should be able to authenticate itself to a remote entity. For example, a corporate network administrator should be able to verify that all machines on his network are running the latest security patches and virus checker files

2.3.3.4.1.5 (U) Implementing Trusted Applications

(U) There are a number of factors that go into the implementation of a trusted application. These include how the application fits into the overall architecture of the system and how it is supported; the development process for the trusted application; and evaluation of the trusted application.

2.3.3.4.1.6 (U) Architecture

(U//FOUO) As noted above, trusted applications must rely on their host platforms to provide some level of security. At a minimum, the application must rely on the host platform to prevent a direct attack on and subversion of the application's security policy. Therefore, implementers of trusted applications must first decide on what platforms the application is to be hosted. Then, they must decide how to use the security features and security policy enforcement provided by the host platform.

(U//FOUO) Most application developers want their software to be able run on a variety of hardware platforms. That maximizes the return on the investment of application development. However, from a security standpoint, that creates a complex situation, since the application developer must decide which characteristics of the family of host platforms are to be supported and clearly state these.

(U//FOUO) For example, an application developer writing a trusted application to run on a server operating system can require that the server and its environment provide identification and authentication of users, tamper-resistance, and a certain level of functionality. Then, any server and environment that provide those features would be an acceptable host for the application.

(U//FOUO) Or, the developer can write the application to require its own identification and authentication service and not rely on an underlying server operating system to provide that feature. This can mean that the application could run on a wider variety of platforms, but at the potential negative of being less friendly to the user (because it would require a separate log-in step).

(U//FOUO) Some applications will be written for use on special purpose devices, such as NSA-certified Type 1 encryption devices. These applications can be tailored for the capabilities of the device to make use of all the features provided by that host platform.

(U//FOUO) Once a platform (or set of platforms) has been selected, the developer must next decide on what features of the platform—and specifically, what security policy features—to rely on to protect the application. For example, an application running on an Multi-Level Security (MLS) or Multiple Independent Levels of Security (MILS) platform may operate at one level or it may support multiple security levels itself. As another example, an application may make use of a platform's strong identification and authentication service by accepting the user identity preferred by the host platform, or it may choose to require its own identification and authentication process.

(U//FOUO) Once all these decisions are made, the developer of the trusted application will know what is the security policy of the application, what parts of the host platform will be used, and therefore what the requirements are for the application itself. The developer can then build to those requirements.

2.3.3.4.1.7 (U) Development

(U//FOUO) A trusted application implements a security policy, even if that security policy is as simple as one that does not contain malicious code. The application must therefore be developed in such a way as to provide some level of assurance that the application does indeed enforce its security policy.

(U//FOUO) The specific development environment and developmental requirements chosen by the developer will be a function of the target assurance level selected by the developer. Typically, the assurance level will be one of the seven Evaluated Assurance Levels (EAL), defined in Volume 3 of the Common Criteria [ISO15408].[10] In this case, the development process must be consistent with the requirements defined for that EAL.

[10] (U//FOUO) This is not required; the assurance level of an application can be whatever level is selected as appropriate for the target environment. However, the U.S. Government approved standard for assurance levels consists of the seven EALs defined in Volume 3 of ISO 15408.

2.3.3.4.1.8 (U) Evaluation

(U//FOUO) Many commercial products support the notion of a trusted application. However, in most cases, a trusted application is simply one that the user decides to trust. Sometimes, the user is told that there is an organization that digitally signed the application, so there is some level of confidence that it came from a particular commercially-reliable organization[11]. There are known weaknesses in systems that rely on digitally signing code to assure its benign properties, and this mechanism by itself is insufficient for the GIG.

(U//FOUO) In short, for GIG users, there is generally no reason for the user to believe that any particular application will enforce its security policy or will not contain malicious logic. The way to improve this situation is to have security-critical applications be evaluated. The Common Evaluation Methodology (CEM), defined under the NIAP, should be used as the baseline. Protection profiles should be defined for any common applications, in the way that [Dprof] and [Wprof] are being defined for database management systems and web servers, respectively. If an application is sufficiently unique to make a protection profile not worthwhile, then an equivalent evaluation should be done based on a security target established for that application.

2.3.3.4.2 (U) Usage Considerations

2.3.3.4.2.1 (U) Implementation Issues

(U//FOUO) The major implementation issues have been described above. They are:

- (U//FOUO) Determining the security policy to be enforced by the application
- (U//FOUO) Determining the set of host platforms on which the application is to be executed
- (U//FOUO) Determining the security policies/properties enforced by those platforms and deciding which of them will be used by the applications
- (U//FOUO) Finally, determining the requirements to be met by the application itself to enforce the selected security policy in the assumed environment

(U//FOUO) The development environment must reflect the chosen assurance level for the application. If required, an independent evaluation of the implemented application must be performed to achieve the required confidence that it enforces its security policy.

2.3.3.4.2.2 (U) Advantages

(U//FOUO) The advantage of a trusted application over a generic, untrusted application is that GIG users and designers have an established level of confidence (the assurance level) that the trusted application enforces its security policy in its presumed environment. This allows users and security officers to better understand the total risks involved in operating the GIG and also improves the security of the GIG.

[11] (U) The "worth" of such a signature, if any, in the commercial environment is that there is some organization that can be sued if the application turns out to damage the user's environment.

(U//FOUO) A trusted application that enforces a complex security policy, such as support for MLS or MILS, can allow enhanced operations for certain GIG users. For example, an MLS e-mail system can allow a user to communicate via e-mail with multiple communities operating at different classification levels simultaneously, eliminating the need for multiple e-mail devices and connections.

2.3.3.4.2.3 (U) Risks/Threats/Attacks

(U//FOUO) Attacks against trusted applications can come either directly against the application itself or through the underlying host platform. This is why the application developer must consider the intended host platform and the totality of the system in designing and implementing the application.

(U//FOUO) An example of an attack against the application itself is feeding the application bad data. Early web servers did not filter data passed to them by clients, and it was often possible to provide a shell script program in the data field of a web form, and thus have the program executed on the web server. Thus it was necessary for web server implementers to filter all data provided by a client to ensure that no malicious logic got executed on the server’s host computer. Application developers must consider similar attacks against their own applications and defend against them appropriately.

(U//FOUO) An example of an attack against an application through the host platform is one in which an attacker places a keystroke logger on a computer to capture all keystrokes typed into an application. This attack can potentially recover passwords, PINs needed to unlock and use private keys, and other sensitive material without the application itself ever being aware of this. Application developers must be aware of the types of host platforms on which their applications will run and the potential attacks that can occur through those host platforms. They must make decisions about which types of attacks to defend against, and which types of attacks to simply accept. The application's documentation must make clear the decisions made by developers and the resultant risks to application users.

2.3.3.4.3 (U) Maturity Level

(U) The maturity level of trusted applications varies according to the type of application and host platform. Trusted applications that enforce simple security policies (e.g., within a single security level) have existed for several years. Standards and guidelines for developing trusted applications such as web servers, multi-level e-mail, and multi-level database management systems (DBMS) exist now or are in development. Some implementations of applications that comply with those standards and guidelines are in various stages of development.

(U//FOUO) However, there is much work to do in the general field of trusted applications. Applications that work across security domains or levels; that enforce dynamic access control policies such as RAdAC; and that work across a range of general-purpose host platforms while communicating across a variety of networks, require significant amounts of research and development.

(U//FOUO) In terms of timelines, the set of trusted applications that are available will increase gradually over time. Simple trusted applications exist today, and there will be a few more by 2008. Self-protecting trusted applications and support for some more complex security policies will exist by 2012. Full support for complex security policies on a variety of host platforms will not exist until the 2016–2020 timeframe.

(U//FOUO) The technology readiness level group assigned for trusted applications is Emerging (TRLs 4 – 6). This is an accurate reflection of the overall status of the area. As noted above, some parts of this field are already well understood, and trusted applications exist. Other areas require significant research and development.

2.3.3.4.4 (U) Standards

(U) Applicable standards for trusted applications are Protection Profiles developed against ISO 15408, the Common Criteria. Because of the different security requirements, security policies, and functional requirements of different applications, it is not possible to have a generic Trusted Application Protection Profile. Rather, a different Protection Profile will need to be developed for each type of application. It may be necessary to have multiple Protection Profiles for some types of applications, depending on the possible security policies that can be assigned for that application.

(U) At the time of this writing, there were no U.S. Government validated Protection Profiles for trusted applications. There are two draft profiles currently being validated, one for database management systems [DBProf] and one for web servers [WSProf].

2.3.3.4.5 (U) Cost/Limitations

(U//FOUO) The costs of trusted applications are associated with the processes that must be followed, particularly the development and evaluation processes. Trusted applications have the potential of being very expensive, particularly if custom development processes must be followed in order to achieve acceptable assurance levels. A goal is to improve the software development process so that standard commercial best practices are sufficient to develop high assurance trusted applications.

(U//FOUO) If trusted applications must be evaluated, the costs of the evaluation are also a concern. A robust, efficient evaluation process must be developed.

2.3.3.4.6 (U) Dependencies

(U) Successful development of robust trusted applications with complex security policies depends on the completion or establishment of:

- (U//FOUO) Dynamic access control policies
- (U//FOUO) Standards for application development and evaluation
- (U//FOUO) Understanding of the relationship between host platform security policies and trusted application security policies
- (U//FOUO) Establishment of techniques and uniform requirements for self-protecting

applications

2.3.3.4.7 (U) Alternatives

(U//FOUO) The only real alternative to trusted applications is to regard all applications as untrusted and rely upon the host platform to provide protection. Depending on the need to share information within a COI, untrusted applications constrained by host platforms may not provide sufficient functionality to accomplish a mission.

2.3.3.4.8 (U) Complementary Techniques

(U//FOUO) Trusted applications work more efficiently with trusted platforms, as that enhances the uses for trusted applications (e.g., MLS e-mail programs on MLS or MILS platforms provide greater functionality than MLS e-mail programs on single-level platforms). In addition, there is greater overall confidence in the security provided by a trusted application running on a trusted platform than there is in a trusted application running on an untrusted platform, because there is less likelihood of an attack on the application coming through the host platform.

2.3.3.4.9 (U) References

(U) [DBProf] – National Information Assurance Partnership, U.S. Government Protection Profile Database Management Systems for Basic Robustness Environments, Version 0.24, 15 December 2003. Available at http://www.niap.nist.gov/pp/draft_pps/pp_draft_dbms_br_v0.24.pdf

(U) [ISO 15408] – a three volume set, consisting of:

(U) ISO 15408-1: Information technology -- Security techniques -- Evaluation criteria for IT security -- Part 1: Introduction and general model, 1999.

(U) ISO 15408-2: Information technology -- Security techniques -- Evaluation criteria for IT security -- Part 2: Security functional requirements, 1999.

(U) ISO 15408-3: Information technology -- Security techniques -- Evaluation criteria for IT security -- Part 3: Security assurance requirements, 1999.

(U) [PEINADO] - Peinado, Marcus, Yuqun Chen, Paul England, and John Manferdelli, NGSCB: A Trusted Open System, Microsoft White Paper, Microsoft Corporation, Redmond, WA, available at http://research.microsoft.com/~yuqunc/papers/ngscb.pdf

(U) Shirey, R., Internet Security Glossary, RFC 2828, May 2000. Available at http://www.ietf.org/rfc/rfc2828.txt

(U) [TDI] – National Computer Security Center, Trusted Database Management System Interpretation of the Trusted Computer System Evaluation Criteria, NCSC-TG-021, April 1991. Available at http://www.radium.ncsc.mil/tpep/library/rainbow/NCSC-TG-021.html

(U) [WSProf] - National Information Assurance Partnership, U.S. Government Protection Profile Web Server for Basic Robustness Environments, Version 0.41, 1 August 2003. Available at http://www.niap.nist.gov/pp/draft_pps/pp_draft_websrv_br_v0.41.pdf

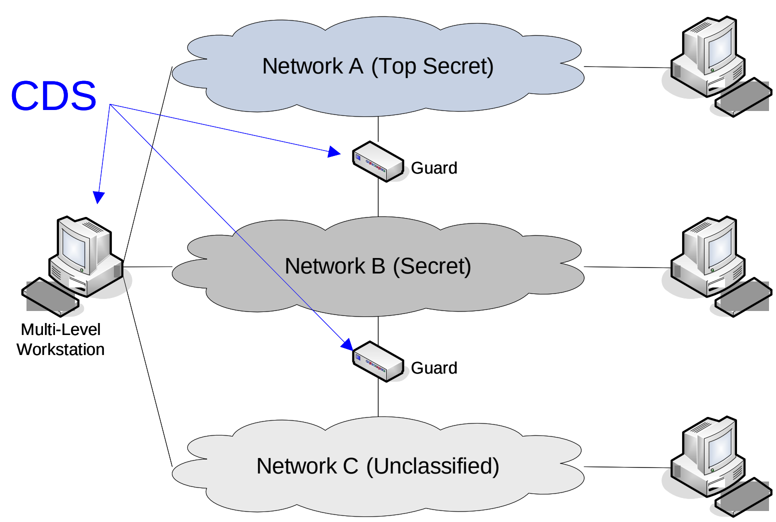
2.3.3.5 (U) Cross Domain Solution Technologies

(U//FOUO) The demands of modern warfare require that deployed forces must tightly synchronize activities in real-time within Joint and international Combined Force environments. Such synchronization necessitates an assured sharing environment where information travels seamlessly across space, time, security, and releasability domains so that the right information gets to the right warfighters in a time and place that maximizes operational effectiveness. The operational need for cross-domain information flow has been recognized for decades. However, the advent of high-speed information systems, their supporting networks, and the subsequent reliance of U.S. and multinational forces on these standardized, high-performance technologies emphasizes an urgent need for assured cross-domain solutions if organizations are to keep pace with doctrinal and operational shifts toward network-centric warfare.

(U//FOUO) A cross-domain solution (CDS) is an information assurance solution that provides the ability to manually or automatically access or transfer data between two or more differing security domains (CJCSI 6211.01b). These solutions should enable the secure transfer of information across differing security domains to sustain and maximize operational effectiveness while supporting GIG security objectives. This document recognizes that while the interconnection of information systems of different security domains within and at the periphery of the GIG may be necessary to meet essential mission requirements, such connections pose significant security concerns and shall be used only to meet compelling operational requirements, not operational convenience (DoD Instruction 8540.aa (DRAFT)).

2.3.3.5.1 (U) Technical Detail

(U//FOUO) The broad definition of CDS given in CJCSI 6211.01b manifests itself in legacy systems as two distinct sets of technologies as shown in Figure 2.3-22.



This figure is (U//FOUO)

Figure 2.3-22 (U) Legacy Manifestation of Cross-Domain Solutions

(U//FOUO) The first set of technologies falls within the category of controlled interfaces or guards. These technologies enable the flow of information between security domains. The second set of technologies deal with accessing multiple domains from a single node, workstation, or server. As technology matures within each GIG IA increment, the distinction between these two sets of technologies will blur substantially.

(U//FOUO) A CDS that is used for the transfer of information could be a device or a group of devices that mediate controlled transfers of information across security boundaries (e.g., between security domain A and security domain B). In this usage, a CDS is trusted to allow sharing of data across boundaries and enforces a defined security policy. CDS take into account the following characteristics: type of data flow, direction of data flow (e.g., high to low, low to high), human or automated review, connection protocol, number of connections) as shown in Figure 2.3-23. Some services of a CDS include filtering, dirty word search, integrity checks, sanitization, downgrading, and virus scanning.

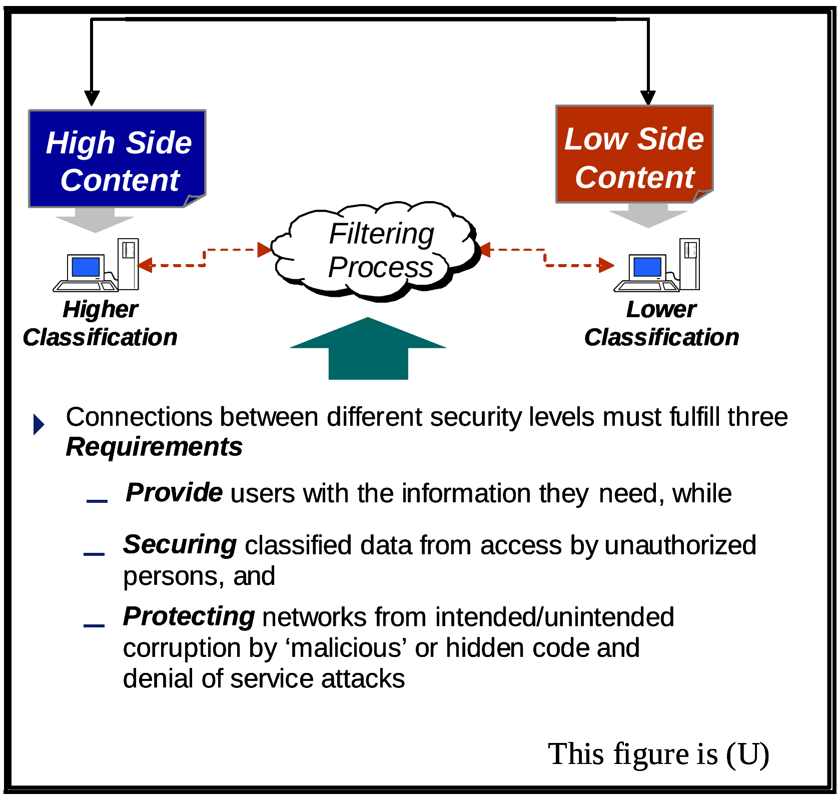


Figure 2.3-23: (U) Controlled Interface Example

(U//FOUO) Current CDS technologies that deal with simultaneously accessing information from multiple domains from a single location typically fall within two categories based on functionality. The first category includes information systems that internally separate multiple single levels (MSL) of security. Two dominant architectures supporting MSL technologies include systems where separation is maintained locally within an edge platform (e.g., thick client architectures such as NetTop), and systems where separation is performed remotely at a server and clients lacking local storage access to each domain as allowed by the server (e.g., thin client architectures such as Multi-Level Thin Client (MLTC). These two MSL architectures, shown in Figure 2.3-24, are complementary and address information access requirements of different operational environments.

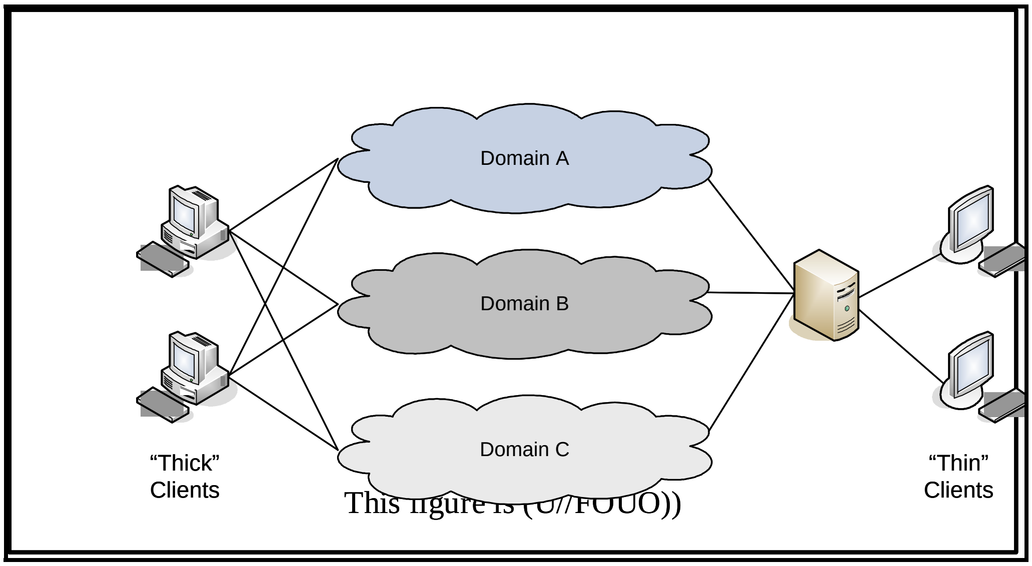


Figure 2.3-24: (U) Two MSL Architectures

(U//FOUO) The second category of information-access CDS includes information systems that can manage multiple levels of security (MLS) simultaneously allowing, for example, cut-and-paste between windows of a MLS workstation. An MLS database, for example, could allow users in different security domains to access information in the same database up to their respective clearance levels (versus accessing information in two distinct replicated databases, one database image within each security domain).

2.3.3.5.2 (U) Usage Considerations

(U//FOUO) Traditional MLS architectures meet many of the requirements of CDS and have been used successfully in limited contexts in the past. The main factors limiting broader acceptance and deployment of MLS solutions historically have been cost and certification and accreditation (C&A) issues with respect to mainstream operating systems and COTS applications in widespread use throughout the DoD (e.g., Windows NT, Microsoft Office). While certain systems (e.g., Wang XTS-300 Trusted Computer System) have been approved by NSA for MLS processing in particular contexts, such systems have historically not supported a broad enough variety of COTS applications and DoD-specific applications to form a viable basis for developing a complete CDS capability. An example of an MLS solution deployed today is OSIS Evolutionary Development (OED).

(U//FOUO) OED has an impressive accreditation and usage record in the DoD. However, there still remain issues that affect its applicability as a general MLS capability. In brief, the issues relate to vendor support for the underlying operating system, support for commercial off-the-shelf (COTS) applications (e.g., MS Office), the ability of serial links used to scale in large environments, the need to operate the system in a Sensitive Compartmented Information Facility (SCIF) with only TS-cleared personnel, and the need to run customized versions of command and control applications.

(U//FOUO) The absence of fully general MLS solutions deployable on a mass scale has led to the proliferation of MSL technologies. With the exception of guard components, MSL solutions are more technically tractable and are often more straightforward to certify and accredit than general purpose MLS technologies. Examples of such solutions deployed today include Coalition Operational Wide Area Networks (COWANs) and, more recently, Combined Enterprise Regional Information Exchange System (CENTRIXS). Additional examples of systems using a MSL architecture are MLTC and Network on a Desktop (NetTop).

(U//FOUO) Experience has shown that MSL architectures have proven unsatisfactory in many settings. MSL often reinforces the dependence on duplicated infrastructures—one for each security level. Duplicate infrastructure exacerbates the multiple sign-on problem (the warfighter must logon to each infrastructure separately rather than using an SSO login) and often leads to increased Space Weight and Power (SWaP). SWaP is particularly acute in many constrained, warfighting environments (e.g., ships, submarines, aircraft, ground vehicles, as well as the hauling capacity of individual troops). While certain approaches (e.g., MLTC and Keyboard, Video, Mouse [KVM] switches) have partially ameliorated the duplicate hardware issue, a number of additional drawbacks remain in MSL.

(U//FOUO) At the most basic level, MSL tends to impede the efficient and timely dissemination of information. The warfighter is hampered with sometimes awkward, frequently insufficient, and at times even inappropriate workarounds. The warfighter must manually switch between disparate security domains and their associated separate databases and networks in order to accomplish the assigned mission. Correlating information between domains is challenging and may result in a warfighter having to resort to various methods, such as rekeying of data, sneaker net, and hard drive swapping, to ferry information between segregated networks. Such procedures introduce risks and delays.

(U//FOUO) Such inefficiencies and risks manifest themselves in many ways from the merely inconvenient to the potentially serious. The result is that MSL drawbacks extend across many areas. Operational shortcomings include situational awareness timeliness and accuracy, situational awareness confusion, data under-classification, data over-classification, information inaccessibility, online searching difficulties, and timeliness of setup for ad-doc coalitions. MSL has a number of technical shortcomings, including guard proliferation breaking end-to-end security assumptions, lack of need-to-know enforcement, identity maintenance and correlation difficulties, collaboration difficulties, timely setup for ad-hoc coalitions, and proliferation of network interconnections.

2.3.3.5.2.1 (U) Implementation Issues

(U//FOUO) CDSs are software/hardware implementations of networked IT. The variability and complexity of IT systems, for practical purposes, result in solutions that have an infinite number of possible configurations. Resources at all levels do not reasonably exist to adequately test and evaluate all possible configurations of CDSs and their component devices. For this reason, configuration of CDSs must be strictly controlled and enforced. Any configuration change outside of that specified could dramatically affect one or more of the three assessment areas mentioned earlier. In addition, CDSs are designed for specific purposes and to handle specific data requirements. Any changes in operational concept or data encoding will most often defeat the security provided by the CDS.

(U//FOUO) Beyond configuration, control, and maintenance, there are fundamental issues that must be addressed when considering the use of CDS within an environment that aims for end-to-end security. Of principle concern is the following. A primary function of CDS is to examine content, and this instantiates numerous conflicts with technologies aimed at protecting confidentiality and integrity (e.g., FNBDT, SSL/TLS, IPsec, and HAIPE).

2.3.3.5.2.2 (U) Advantages

(U//FOUO) Until the GIG 2020 Vision is achieved, multiple security domains will exist. CDS is essential to allow information sharing among GIG entities during this transition period. CDS will remain a necessary component of interoperability within multinational forces whose technology procurement schedules are not dictated by the GIG acquisition timelines.

2.3.3.5.2.3 (U) Risks/Threats/Attacks

(U//FOUO) Any use of a CDS entails an acceptance of risk. Risks exist for both the inadvertent release of restricted data as well as the risk of malicious attack. For community-operated networks, the risk assumed by a CDS is imposed on all network operations and is not restricted to the specific system requiring the CDS. All CDSs represent some level of risk, and a CDS should not be contemplated except under compelling operational requirements. In considering a CDS for use, the specific and community risks must both be assessed before any accreditation decision is made.

(U//FOUO) The risk of a CDS is comprised of more than just the connection technology. It must encompass the data/application environment and risk posture of the connecting enclave as well. The three assessment areas required for any CDS are:

- (U//FOUO) Connection Confidence – An assessment of how confident the solution will behave as specified and is resistant to exploitation.
- (U//FOUO) Data Potential – An assessment of the volatility of the data formats/types allowed by the connection and their potential to cause harm in the operational environment.
- (U//FOUO) Partner Type – An assessment of how likely the connection system or its administrator would support/sponsor an attack.

(U//FOUO) It is important that cross-domain solutions be understood to be holes intentionally placed within a more strict security environment for the purpose of improved information sharing. Current CDS technologies provide no ability to mitigate filtering or disclosure errors.

2.3.3.5.3 (U) Maturity

(U//FOUO) This section describes CDS that currently exist, new cross-domain solutions being considered for near term development, and systems that will require research and the use of future technologies.

2.3.3.5.3.1 (U) Current Technologies and Solutions

(U//FOUO) Currently, most operational cross-domain solutions fall into one of four technology areas. These areas are electronic mail (e-mail), fixed formatted data, file transfer, and desktop reduction. The lack of maturity of underlying IA controls causes these technologies to be considered Emerging (TRLs 5 - 7), even though many of these technologies have been demonstrated in an operational environment.

(U//FOUO) E-mail cross-domain solutions scan ASCII-formatted e-mail for dirty words as messages traverse from high-side (e.g., SIPRNet) Mail Transfer Agents (MTAs) to low-side (e.g., NIPRNet) MTAs, helping prevent the disclosure of classified information. For low-to-high data flows, e-mail solutions check e-mail for malicious code. As an additional mitigation, senders and recipients can be restricted to a list of those permitted to pass messages through a particular e-mail solution. E-mail attachments are allowed to traverse security boundaries in some cases, but the file types are limited. The majority of e-mail solutions consist of the Defense Information Infrastructure (DII) Guard.

(U//FOUO) Fixed format cross domain solutions transmit ASCII data that conforms to pre-defined format requirements (such as field length, allowed characters, numerical ranges). The strict formatting requirements applied to data submitted for traversal of the security boundary act as a mitigation of the concerns with unintended release and malicious content. The two main solutions that meet the needs of this category are Defense Information Systems Agency's (DISA) Command and Control Guard (C2G) and the Radiant Mercury (RM).

(U//FOUO) File transfer solutions allow data files to be transmitted across the boundaries of their original security level. The files allowed to pass through the solution currently are only those considered low-risk data. This is due to the complexities of many file formats. Therefore, most solutions only allow the passage of plain ASCII text documents and image files. Additionally, high-to-low flows require human review prior to release to the low side. Currently the Imagery Support Server Environment (ISSE) Guard and the Trusted Gateway Solution (TGS) are the two solutions used most frequently for this type of data transfer.

(U//FOUO) Desktop Reduction is a valid concern in business today. The need to have access to multiple networks of different security domains in one location is a necessity in many environments. The problem faced here is the user now has multiple desktop computers using up his/her desk space. In the cases of small office space or aboard a ship, space is at a premium. The idea of desktop reduction is to free physical workspace and decrease the footprint of the computers. In the example of a KVM switch, the user only requires one monitor, one keyboard, and one mouse. In other solutions presented, the user may only require one desktop computer and one monitor to access these multiple networks.

2.3.3.5.3.2 (U) New Cross-Domain Solutions Being Considered for Near-Term Development

(U//FOUO) Many technologies fit into this category including chat, file transfer (high risk data), Browse Down, and Content-Based Information Security (CBIS). These technologies are considered Early/Emerging (TRLs 3 - 4).

(U//FOUO) Chat is a technology most people are now familiar with. Many commercial Instant Messengers can be downloaded free today from the Internet. Chat gives the user the ability to communicate with co-workers and friends in real-time by sending text. In the cross-domain world, Chat would allow a user to send text across security domains in near-real-time (allowing some latency for filtering).

(U//FOUO) In the world of Cross-Domain, file transfer has always been a big issue. Although accredited solutions exist to transfer fixed file formats, there are many files prohibited from being passed through these solutions (e.g., executable files, documents with macros). The technology exists currently to filter some of these higher risk data types. One of the larger pieces of the puzzle lies in filtering Microsoft Office documents since they are so widely used. Microsoft Office documents are considered to be high risk due to all of the hidden information and executable contents (macros) which can be stored in them. With the right solution in place, business could carry on relatively seamlessly between security domains.

(U//FOUO) Browse Down is a technology used to browse a lower security domain from a higher security domain network. One example of this would be to surf the Internet while attached to your classified network at the office. This would alleviate the need to purchase more hardware for the user’s workspace and pull a network feed to his/her desk.

(U//FOUO) CBIS is the direction many in the Cross-Domain community are going. CBIS can provide controlled access to assets based on the attributes associated with them. These attributes will include a security classification as well as a need-to-know attribute. CBIS is policy driven, which dictates a specific role to a user. After using strong Identification and Authentication (I&A) mechanisms to help enforce the access control, a user is only permitted to access files, which his/her role allows them to see. Although some of this technology exists today, it is relatively in its infancy. When this technology has matured, central repositories will be able to hold information from multiple security domains and allow CBIS to drive the policy. It is anticipated that the key Assured Information Sharing technologies developed by CBIS will be incorporated into other cross domain solutions.

2.3.3.5.3.3 (U) Future Technology and Research Needed

(U//FOUO) In general, for any mission-essential IT services within a system (security domain) a requirement will exist for that IT service to be supported across systems. Today, key IT services are e-mail, sharing files, collaboration, and web browsing. In the future key IT services are expected to include XML-based Web Services, VoIP, and others. In addition, the 2020 Vision is for a single system that can support as many security domains as needed. Given the diversity of these requirements for CDSs, to date research and development has provided solutions to only a small portion of the requirements, and for those requirements that can be satisfied with CDSs today the overall administration of the CDSs is very labor intensive. Several areas for research and development exist that would target making existing CDSs more enterprise-enabled and net-centric. The objective is to have near-term return on investment by enhancing the collaboration capabilities supported by CDS, bringing existing CDSs into compliance with standards and necessary assurance levels, and making their administration less labor intensive.

(U//FOUO) Research and development is also needed to address cross-domain security issues for particular capabilities operating within an environment supporting end-to-end security. For example, research and development is needed to address the cross-domain security issues with VoIP within an environment supported by HAIPE. Likewise this applies to Web Services, and for collaboration capabilities such as virtual white boarding, shared applications, remote desktop control, audio/video conferencing, etc. This research and development will address gaps in our knowledge of how to architect cross-domain capabilities into the GIG vision.

(U//FOUO) As the GIG evolves towards the 2020 Vision, CDSs as we often see them today (devices at the system boundary) will continue to evolve and exist, primarily to control the flow of information between the GIG and non-GIG systems, such as the Internet, coalition networks owned by multiple nations, national networks owned by another nation, and possibly other U.S. Government agencies. Of course, CDSs controlling information passing into and from the GIG will need to be GIG-compliant and net-centric themselves.

(U//FOUO) Achieving the 2020 Vision of a single system capable of handling all types of DoD information will require that virtually all components within the GIG incorporate, to some extent, the techniques and technologies first developed and deployed at the boundaries in CDSs.

(U//FOUO) For example, as we pursue research and development to establish a capability to examine Microsoft Office files for executable and hidden content, that capability will likely first appear in a CDS. Initial capabilities such as this are often complex and costly, making them unwieldy for initial deployment on every desktop. Instead, these complex and costly capabilities will appear in centralized locations that are already—basically—complex and costly, where application of the capability/checking can be assured, and where the benefit of the capability has the largest payoff. As the capability matures, it would migrate from the CDSs to the desktops themselves. To achieve the 2020 Vision, a capability such as in this example, will likely be a key requirement for protecting the GIG's confidentiality, integrity, and availability.

(U//FOUO) Another example would be the ability to discern the meaning of a document from its content. While this capability is costly and complex, it will likely be used in a CDS to detect and prevent inappropriate information from being released/disclosed. As the capability matures, it can migrate to the desktop so that in 2020 a person preparing a document for a given recipient will receive a flag/notice if the tool determines from the content of the document that it would be inappropriate for that intended recipient. These are examples of how techniques and technologies originally implemented in CDSs can mature and then migrate from the boundary of the GIG into components within the GIG, and thus are critical enablers to achieving the GIG 2020 Vision.

2.3.3.5.4 (U) Standards

(U//FOUO) Standards for addressing Cross Domain requirements listed in Table 2.3-12.

Table 2.3-12: (U) CDS Standards

This table is (U//FOUO)		
Name	**Description**	**Applicability**
CJCSI 6211.02B	Defense Information System Network (DISN): Policy, Responsibilities, and Procedures	This Instruction applies to the Joint Staff, Combatant Commands, Services, Defense Agencies, DoD Field Activities, and Joint Activities. It addresses Cross Domain requirements and the policy, responsibilities, and procedures for resolving Cross Domain issues. Cross Domain connections shall be in accordance with DoD Directive 8500.1, Information Assurance, and DoD Instruction 8500.2, Information Assurance Implementation. Procedures within DoD Instruction 5200.40, DoD Information Technology Security Certification and Accreditation Process, including a risk assessment by the Cross Domain Technical Advisory Board and approval by the DISN Security Accreditation Working Group, will be followed.
DCID6/3	Protecting Sensitive Compartmented Information within Information Systems	This is a mandate for the Intelligence Community (IC). It is not applicable within the DoD unless a DoD system is connected to an IC system. The Policy portion of this Directive establishes security policy and procedures for storing, processing, and communicating classified intelligence information in information systems (ISs). It lists policies plus roles and responsibilities. The Manual portion of this Directive provides policy, guidance, and requirements for ensuring adequate protection of intelligence information. It includes a section on Controlled Interfaces, which are used for interconnected ISs, including those of different security domains.

This table is (U//FOUO)

Name	Description	Applicability
(DRAFT) DODI 8540.aa		Interconnection and Data Transfer between Security Domains. This Instruction will establish the DoD policy, responsibilities, and procedures for Cross Domain interconnections and the engineering, installation, certification, accreditation, and maintenance of such interconnections. Upon publication, it will apply to the Office of the Secretary of Defense, Military Departments, Chairman Joint Chiefs of Staff, Combatant Commands, Inspector General of the DoD, Defense Agencies, and DoD Field Activities. It applies to all DoD information systems. It includes Cross Domain Connection Request Procedures, a Cross Domain Data Transfer Generic Framework and Scenario, and Controlled Interface Characteristics.

This table is (U//FOUO)

2.3.3.5.5 (U) Cost/Limitations

(U//FOUO) Cross domain solutions are Government Off-The-Shelf (GOTS) products because they require higher assurance levels than available commercially.

2.3.3.5.6 (U) Dependencies

(U//FOUO) Advancement of CDS technologies is heavily dependent upon the development and management of trusted platforms and trusted applications. The success of CDS technology in enhancing operational effectiveness depends substantially on the involvement of Programs of Record in developing collaboration tools as well as command and control applications that are CDS aware. The ability of CDS to enhance force protection capabilities, avoidance of blue-on-blue engagements, and rapid dissemination of blue force indications and warnings depends heavily on our ability to put CDS-aware capabilities directly in the hands, cockpits, and workspaces of our warfighters.

2.3.3.6 (U) Non-Repudiation

2.3.3.6.1 (U) Technical Detail

2.3.3.6.1.1 (U) What is Non-Repudiation

(U//FOUO) Non-repudiation is a service used to protect against an entity that attempts to falsely deny, such as falsely denying generating a message or falsely denying receipt of information. Strictly speaking, technical non-repudiation mechanisms cannot actually prevent an entity from denying participating in an action or communication. Instead, they provide evidence that can be used to refute the repudiation claim. That is, the goal of a non-repudiation service is to provide a presumption that the entity performed the action in question and force the entity to provide strong evidence that it did not.

(U) An example is in order. Suppose that Alice and Bob are in business. Bob presents a purchase request purported to be from Alice and asserts that Alice thus owes him money. However, Bob could well be forging the request himself, or it could have come from another entity entirely. It could have actually come from Alice, who now wants to disclaim it, as she does not wish to pay the money owed. A non-repudiation service would allow Bob to go to a neutral third party—such as a court—and convince it that Alice really did send the purchase request. Conversely, it would provide Alice strong proof that she did not send the purchase request.

(U) As defined in the International Standards Organization's Open Systems Interconnection Reference Model (ISO/OSI 7498 part 2), there are two basic types of non-repudiation service:

- (U) Non-repudiation with proof of origin: A security service that provides the recipient of data with evidence that can be retained and that proves the origin of the data, and thus protects the recipient against any subsequent attempt by the originator to falsely deny sending the data. This service can be viewed as a stronger version of a data origin authentication service, because it can verify identity to a third party

- (U) Non-repudiation with proof of receipt: A security service that provides the originator of data with evidence that can be retained and that proves the data was received as addressed. This thus protects the originator against a subsequent attempt by the recipient to falsely deny receiving the data.

(U/FOUO) These two services both deal with network communications, that is, the sending and receipt of a message. In the GIG, the concept of non-repudiation must be generalized to address a variety of other actions, such as over-riding security policies, granting access to classified information to entities without appropriate clearance, etc.

(U) Non-repudiation has both technical and non-technical components. The technical measures involved in a non-repudiation service include:

- (U) Authentication of the identity or identities associated with a transaction or transmission. The authentication MUST be such that, with a very high degree of confidence, only one entity can provide the correct authentication information. Typically, this is done by the use of a PKI, where each entity is assigned a private key to use for authentication/digital signature, and this key is not determinable by any attacker—given assumed efforts

- (U) Integrity of the information. Once an entity has taken some action—sent or received a message; taken part in a transaction—it must not be possible for any attacker to modify the contents/records of that transaction. Typically, this is accomplished using digital signatures—the entity signs the message/transaction/ record, and any modification to that signature or record is detectable

- (U) Time Stamping. One of the problems with signature-based systems is that back-dating of records/events is possible. Suppose that Alice has a private key used for digital signatures. If Alice's key is compromised for whatever reason (e.g., she loses the token on which it is stored, along with the PIN to that token), an attacker (Mal) who now knows the private key can create various records and assign to them whatever time Mal desires. Mal can create signed records that are dated before the compromise occurred—even years earlier, if desired. To protect against this, a third-party time stamping service can be used, to indicate that a record did exist no later than a given time. Any records presented without time stamps are not considered to be protected by the non-repudiation service

(U) Notarization. Even stronger than a time-stamping service is a digital notarization service. With this service, an entire transaction is certified and recorded by a neutral third party. This provides a stronger chain of evidence for the transaction.

(U) As noted, there are both technical and non-technical components of a non-repudiation service, and no technical service can ever prevent an entity from attempting to deny, or repudiate, an action. Some of the grounds for denial or repudiation could include:

- (U) Compromise of the key. If the authentication service is provided by means of a PKI, Alice can claim that her key was compromised (e.g., stolen), and she did not know it. Thus, she is not responsible for the transaction

- (U) Weakness of the PKI. Alice can attempt to claim that her private key was known to attackers due to procedural or technical weaknesses in the PKI itself. For example, the cryptography was not strong enough, and thus an attacker figured out her private key; or the key purportedly issued to her was actually given to another entity, etc.

- (U) Intent. Alice can claim that the transaction in question was not the one in which she intended to participate. For example, a worm program modified the data; what she saw on her computer screen is not the same as what is in the message. Or, an attacker broke into her computer and used her private key to sign a message without her knowledge. Or, that she did not understand the nature/content of the transaction; she merely clicked OK when presented with a confusing license agreement on her screen

(U) All of these are within the legal scope of non-repudiation, but are outside the technical scope. To date, there is essentially no case law that exists to guide system designers/evaluators in determining what would happen in each of these situations, and what they should do to defend against them. Thus, any non-repudiation service deployed in the GIG should be regarded as providing technical non-repudiation only and not regarded as providing any basis for the resolution of a legal dispute.

2.3.3.6.1.2 (U) Providing Non-Repudiation

(U//FOUO) In the GIG, non-repudiation is required in conjunction with the TPPU model. The non-repudiation service will be applicable to the source and receipt of posted data.

(U//FOUO) Trust of GIG data is associated with the source of the data, particularly since a large number of users may post data of varying confidence. Thus, any user of the data must reliably know the source of the data in order to be able to use it effectively. Where proof of source may be needed, non-repudiation should be applied to the data.

(U//FOUO) Traditional application level non-repudiation services should also be available outside the scope of the TPPU model. Certain security critical events will require authorization by a third party. Non-repudiation evidence of the source or the authorizer of the events will be useful for the investigation of security incidents. GIG security policy will identify certain events as security critical. For example, an access that violates traditional mandatory access control may be identified as a security critical event that requires authorization by a third party.

(U//FOUO) There are three steps in the non-repudiation service: (1) a request for the service (either implicit or explicit), (2) the creation and storage of the non-repudiation evidence, and (3) the retrieval of the evidence, either to assess its acceptability for future non-repudiation or to actually refute a repudiation claim. Requests for service are typically handled in the specific application requiring non-repudiation.

(U//FOUO) We will now address the technology requirements for the components involved in creation and storage of non-repudiation evidence.

2.3.3.6.1.2.1 (U) Authentication

(U) A fundamental requirement for non-repudiation is to be able to authenticate the entity involved in the transaction. Authentication helps to ensure that the entity involved is the one expected to be involved.

(U//FOUO) As noted above, a major requirement to achieve non-repudiation is that the authentication process is very strong. If an attacker can successfully authenticate as another entity, then non-repudiation cannot be provided. For this reason, non-repudiation services are typically based on public-key infrastructures. Authentication is based on possession of a token containing a private key, as well as knowledge of the PIN associated with that token, or with one or more specific biometric properties used to unlock the token. Authentication using passwords is not acceptable for non-repudiation systems, as they are too weak and easily defeated. For example, if Bob wishes to repudiate a transaction, he could simply post his password in a public location, and thus show a strong possibility that it was not he involved in the transaction.

(U//FOUO) For the threshold (2008) GIG instantiation, any application requiring a non-repudiation service must require authentication based on a token, such as the DoD Common Access Card (CAC) and with a PIN or biometric property required in association with the token. As this is already available, the threshold GIG should be able to meet its authentication requirements for non-repudiation.

(U//FOUO) Future iterations of the GIG will require stronger versions of the token. For example, the DoD should advance to a Class 5 PKI for tokens to be used for non-repudiation in the objective GIG. See section 2.7 for a description of the issues related to the DoD PKI.

2.3.3.6.1.2.2 (U) Integrity

(U//FOUO) Once a transaction has occurred, or a message has been sent or received, the record of that transaction or message must be preserved. In order for a non-repudiation service to be provided, it must not be possible to modify the transaction from the time it is created, without that modification being detectable. For example, if Alice creates a message saying, “Pay Bob $100.00”, it must not be possible for anyone, including Alice or Bob, to change the message to “Pay Bob $1.0000” or “Pay Bob $10000” or “Pay Fred $100.00” without the change being detectable.

(U//FOUO) This protection against undetected modification is referred to as an integrity service. Integrity is a mandatory requirement for non-repudiation to be provided.

(U//FOUO) Integrity can be provided through a number of different mechanisms. One common mechanism is through a digital signature. The record (transaction, message) is hashed, and then the hash is digitally signed. Anyone, using only publicly available information (e.g., the public signing key, and the hashing/signature algorithms used) can verify that the purported record has not been changed by validating the digital signature on it. If the hash value is different, the record has been changed and must not be regarded as valid. If the digital signature cannot be verified, the association of the record with the purported sender must not be regarded as valid.

(U//FOUO) A second way to provide integrity is to use Message Authentication Codes (MACs) and specifically, Hashed Message Authentication Codes (HMACs). In an HMAC, a shared secret, such as an AES symmetric key, is known by both Alice and Bob, but no one else. The shared secret is added to the record, and then the entire quantity is hashed. The integrity of the message can be validated by anyone who knows the shared secret, simply re-calculate the hash given the purported record. If it validates, the record was created by someone who knows the shared secret; if not, the record has been modified and must not be regarded as valid.

(U//FOUO) Both of these methods have potential weaknesses. In the digital signature method, if the private signature key is compromised, anyone can create a new record saying whatever the attacker wants, hash, and sign it, and it will be accepted as legitimate by anyone validating the record. Possession of a signed record in that case indicates that the record has not been changed since it was generated, but it does not prove anything about who generated the record, or when, nor indeed show that Alice or Bob had anything to do with the record.

(U//FOUO) The HMAC method is vulnerable to compromise of the shared secret (i.e., the symmetric key). If Mal knows the shared AES key used by Alice and Bob, Mal can create whatever records he wants. This prevents anyone from making valid statements about whether Alice or Bob is responsible for a record.

(U//FOUO) In addition, both methods are vulnerable to weaknesses in or attacks against the hash algorithm used. If it is possible to invert a hash (i.e., given a hash value, find a valid message that results in that hash value) then an attacker could create or modify records undetectably.

2.3.3.6.1.2.3 (U) Time-Stamping

(U//FOUO) As noted above, non-repudiation services are vulnerable to after-the-fact attacks, such as the compromise of a private signature key. An attacker, Mal, who learns Alice's private signature key can create records and then back-date them. Mal can, for example, create records indicating Alice promised to pay him money several years ago.

(U//FOUO) This attack is particularly worrisome in the context of non-repudiation, in situations in which Alice may want to repudiate a record. That is, Alice may promise to pay Bob a large sum of money, but then want to back out of the obligation. Alice may even try to deliberately disclose her private signature key. By showing that the key was compromised, she can then cast doubt as to whether she was the real originator of the record, and thus may be able to avoid her obligation.

(U//FOUO) To combat this attack, a system can employ trusted time-stamp and notary services. These services support the non-repudiation service by supplying proof of when information was signed. In a trusted time-stamp service, a neutral but trusted third party creates a record of when some specific record existed. That is, the time-stamping service certifies (e.g., through a digital signature of its own) that a record, R, of Alice's actions existed at time T. Later on, if there is a question about the validity of some action, this record can be consulted. For example, if Alice's private key is compromised, and a record R' is produced that is dated before time T, but there is no time-stamp of R' from the time-stamping service, R' will be rejected as invalid. However, if Alice tries to repudiate her original record R by showing that her private key was compromised, but the time-stamped record existed before the compromise, then the validity of the record would be upheld.

(U//FOUO) In order to provide a time-stamping service, a number of items are needed. First, the time-stamping service needs access to a clock whose accuracy is accepted by all parties. (That is, it should not be possible to manipulate the clock to deliberately set the time ahead or back. Similarly, the clock drift must be acceptably small. The acceptable drift will depend on specific applications—in some cases, millisecond accuracy will be required; in others, it will acceptable if only the day is correct.)

(U//FOUO) Second, the time-stamping service must have a very strong digital signature capability. Typically, this would be a more secure digital signature capability (e.g., longer private key length; more tamper-resistant signing module; higher-assurance procedures) than regular system users.

(U//FOUO) Third, there must be a way to securely store and retrieve time-stamped records. Even if the records cannot be manipulated without detection, no useful service has been provided if they cannot be found and used when needed.

(U) Some work on time-stamping standards and requirements has been done. For example, the IETF has developed RFC 3161, Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP). The European Technical Standards Institute (ETSI) has also developed a number of standards relating to time-stamping; for example see ETSI ES 201 733, "Electronic Signature Formats".

(U//FOUO) The basic technical requirements of time-stamping can be met with current technology, such as PKI-based digital signatures. Improving the service requires a stronger PKI, such as a future, higher-assurance version of the DoD PKI. Other improvements in time-stamping all rely on stronger procedures and personnel security.

2.3.3.6.1.2.4 (U) Notarization

(U) Time-stamping provides third party evidence that a particular record existed at a specific time. A stronger service is digital notarization. Notarization adds to the time-stamping service by generating and preserving authenticating evidence, such as digital signatures, associated X.509 certificates, and related certificate validation data (e.g., a validation path or an On-Line Certificate Status Protocol transcript). The authentication evidence for a record may itself be signed, time-stamped, and stored for future use.

(U//FOUO) Notarization, thus, shows the complete state of the system—to the extent that it was knowable—when a specific record was generated. Notarization not only shows that the record existed at time T, but also that at time T, the certificates used were not compromised or revoked, and that the purported user had been successfully authenticated. Any other relevant system state can also be captured.

(U//FOUO) As with time-stamping, a number of items are needed for a notarization service to be provided. First, it requires time-stamping. Second, the notarization service must have a very strong digital signature capability. Typically, this would be a more secure digital signature capability (e.g., longer private key length; more tamper-resistant signing module; higher-assurance procedures) than regular system users. Third, the notarization service needs reliable access to current system information (e.g., certificates; Certificate Revocation Lists or OCSP responses; authentication information). Finally, there must be a way to securely store and retrieve notarized records.

2.3.3.6.2 (U) Usage Considerations

(U//FOUO) The decision to deploy a non-repudiation service, and what type of service to deploy, will be influenced by a number of factors. These include the costs of service deployment (including system and connectivity costs, as well as costs in terms of the number of people required to install and maintain the service, and the performance costs involved in the actual operations), and the benefits gained by deploying the non-repudiation service.

2.3.3.6.2.1 (U) Implementation Issues

(U//FOUO) There are a number of implementation issues that must be considered when deploying a non-repudiation service. These directly affect the cost, staffing levels, and level of security required.

(U//FOUO) First, the appropriate level of authentication must be selected. A non-repudiation service depends completely on the correct authentication of each entity (e.g., each user, group, process). If the authentication system selected is weak (e.g., user-identifier and passwords), then it will be relatively easy to defeat the non-repudiation service. An attacker can simply guess a user's password, or a user attempting to repudiate an action can simply post his password on a public repository. Stronger authentication systems, such as those based on one-time passwords, hardware tokens, or biometrics, provide better security for a non-repudiation system but are more costly to implement. Authentication systems are described in detail in Section 2.1 of this document.

(U//FOUO) Another issue impacting non-repudiation is key management. Whether symmetric cryptography or public-key approaches are chosen, there must be some way to securely generate the keys/shared secrets, distribute them to the proper users, revoke them when necessary, and in general manage these important data items. Key Management is described in detail in Section 2.7 of the Roadmap.

(U//FOUO) Appropriate time-stamp and notarization services must be deployed, if required. Access to sufficiently accurate clocks must be secured, and servers providing the time stamp and notarization functions must be deployed. Sufficient access (e.g., 24/7 uptime with a minimum response time of X; or whatever other metric is required) to these services must be provided. This will create support, configuration, and maintenance issues.

(U//FOUO) Records storage and retrieval must be provided. The purpose of a non-repudiation service is to be able to prove to a third party, if required, that an entity did or did not participate in some event. Depending on exactly what parameters are chosen, the records must be stored for some period of time, with access available within a given level of time when required, and strong security to protect the records from modification or deletion.

(U//FOUO) The decisions made for each of these issues have implications in the number of people needed to operate the system; the trust and skill level that are required by those people; the degree of access and backup required for the systems that implement the function; and other management aspects. All of these impact the cost of implementing a non-repudiation service, and the strength that that service provides.

2.3.3.6.2.2 (U) Advantages

(U) The biggest single advantage to a non-repudiation service is that, if implemented properly, it can provide a strong level of accountability for individual actions. It will be extremely difficult for an entity to falsely deny participation in some event (e.g., there will be strong records that Bob did access particular data, or sent a message, or received a message).

2.3.3.6.2.3 (U) Risks/Threats/Attacks

(U//FOUO) There are two primary failure modes of a non-repudiation service. One is that an entity can successfully repudiate participation in an event in which that entity really did participate. The other is that an entity can be wrongly blamed for participating in an even in which that entity did not participate.

(U//FOUO) The risks to the non-repudiation service that would allow either of these failure modes to occur have largely been discussed above. They include:

- (U//FOUO) Compromise of a private key or shared secret, allowing attackers to forge or modify records
- (U//FOUO) Failure of authentication mechanisms, allowing an attacker to successfully assume an identity
- (U//FOUO) Failure of the integrity mechanism, allowing undetected modifications to records after they have been created
- (U//FOUO) Failure of the personnel or procedural security mechanisms, allowing attackers access to the system or causing records to not be available for examination when required
- (U//FOUO) Insufficient recording, time-stamping, or notarization services, allowing an entity to successfully repudiate an action by, for example, deliberately compromising a private key or shared secret.

(U//FOUO) The biggest risk to a non-repudiation service at this time is that it will be deemed not sufficient by legal authorities when it is required. This can only be solved by working through a number of cases, and developing a body of case law that shows clearly what is sufficient and what is not sufficient for a true non-repudiation service.

2.3.3.6.3 (U) Maturity Level

(U//FOUO) As noted above, the technical requirements for a robust non-repudiation service can be met today. The issues involved in setting up such a service are mostly legal. There is no legal precedent for what is minimally required or acceptable, and very little indication from the legal community as to what is acceptable. For example, the American Bar Association's Information Security Committee has declined to set standards or make recommendations on what is acceptable under U.S. laws for non-repudiation systems. Technical people, such as system and application developers, are making their best guesses as to requirements. However, under U.S. laws, any entity can always attempt to deny or repudiate any action, and then it becomes a matter for the courts to determine whether the technical measures provided were adequate to prevent a successful false denial. Once a body of case law has been established, it may well be possible to set more concrete technical standards.

(U//FOUO) Non-repudiation technology is considered to be Mature (TRLs 7 – 9). As noted above, the technical solutions are known, although individual technical protections could be strengthened. The major developments needed are in the process and legal arenas.

2.3.3.6.4 (U) Standards

(U) The standards that address the technical measures required to provide a non-repudiation service include the ISO's 3-part standard 13888 and the European Technical Standards Institute's "Electronic Signature Formats" work. Specific references are listed in Table 2.3-13.

Table 2.3-13: (U) Non-Repudiation Standards

This table is (U)	
Name	**Description**
ETSI ES 201 733	European Technical Standards Institute, "Electronic Signature Formats", 2000. Available at http://webapp.etsi.org/exchangefolder/es_201733v010103p.pdf
ISO 13888-1	International Standards Organization, "IT security techniques -- Non-repudiation -- Part 1: General", 2004
ISO 13888-2	International Standards Organization, "Information technology -- Security techniques -- Non-repudiation -- Part 2: Mechanisms using symmetric techniques", 1998
ISO 13888-3	International Standards Organization, "Information technology -- Security techniques -- Non-repudiation -- Part 3: Mechanisms using asymmetric techniques", 1997.
This table is (U)	

2.3.3.6.5 (U) Cost/Limitations

(U//FOUO) As noted in the Implementation Issues section, above, the costs of a non-repudiation system are largely driven by the choices made in how strong the system is to be. Costs can be quite large, if real-time access to stored records from several years ago is required and if solutions are chosen that require highly-trusted system operators with a very high skill level. Other cost factors include the strength of the authentication system and the key management solution required.

(U//FOUO) The single biggest limitation of a non-repudiation system is that an entity can always attempt to deny having done something, and the legal system may or may not accept the evidence provided by the non-repudiation system.

2.3.3.6.6 (U) Dependencies

(U) As noted above, a non-repudiation service depends on the proper implementation of a user authentication service, a data integrity service, and a time-stamping or digital notary service. In addition, non-repudiation depends on system access controls and system integrity, and on the proper enforcement of system procedures and processes to prevent modification or deletion of records.

2.3.3.6.7 (U) Alternatives

(U) There are some alternatives into how a non-repudiation service can be provided. It can be based on digital signatures from a PKI. It can make use of MACs and HMACs. It can use time-stamping, or digital notary services. The strength and robustness of the service needed will determine which choices are needed.

(U) If what is desired is a way of proving to a neutral third party that one or more record is valid, or that an entity did or did not participate in a transaction, there is no alternative to a non-repudiation service.

2.3.3.6.8 (U) Complementary Techniques

(U//FOUO) Non-repudiation systems can be used in combination with existing authentication and accountability systems to provide a stronger level of user accountability. Where the technical measures provided by a non-repudiation service are deemed to be insufficient, they can be combined with stronger procedural requirements of personnel security requirements to provide a system of the necessary strength.

2.3.3.6.9 (U) References

(U) ETSI ES 201 733: European Technical Standards Institute, Electronic Signature Formats, 2000. Available at http://webapp.etsi.org/exchangefolder/es_201733v010103p.pdf

(U) ISO/OSI 7498-2: International Standards Organization, Information processing systems -- Open Systems Interconnection -- Basic Reference Model -- Part 2: Security Architecture, 1989.

(U) ISO 13888-1: International Standards Organization, IT security techniques -- Non-repudiation -- Part 1: General, 2004.

(U) ISO 13888-2: International Standards Organization, Information technology -- Security techniques -- Non-repudiation -- Part 2: Mechanisms using symmetric techniques, 1998.

(U) ISO 13888-3: International Standards Organization, Information technology -- Security techniques -- Non-repudiation -- Part 3: Mechanisms using asymmetric techniques, 1997.

(U) RFC 2104: Krawczyk, H., M. Bellare and R. Canetti, HMAC: Keyed-Hashing for Message Authentication, February 1997. Available at http://www.ietf.org/rfc/rfc2104.txt

(U) RFC 3126: Pinkas, D.; J. Ross and N. Pope, Electronic Signature Formats for long term electronic signatures, September 2001. Available at http://www.ietf.org/rfc/rfc3126.txt

(U) RFC 3161: Adams, C., P. Cain, D. Pinkas and R. Zuccherato, Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP), August 2001. Available at http://www.ietf.org/rfc/rfc3161.txt

2.3.4 (U) Protection of User Information: Gap Analysis

(U//FOUO) Table 2.3-14 is a matrix listing basic requirements for secure voice compared with the secure voice-related technologies described in previous sections. Their adequacy of the technologies to meet the 2008 attributes is shown. Some of the IA attributes do not have RCD capabilities mapped to them because they are below the detail specified in the RCD.

Table 2.3-14: (U//FOUO) Secure Voice Technology Gap Analysis

This Table is (U//FOUO)

		Technology Category					
		FNBDT	Interop / Gateways	FNBDT Voice over IP	VoIP Call Control	IP Encryption	Required Capability (attribute from RCD)
IA Attributes	Type 1 End-user to End-user Confidentiality		N/A		N/A		IAAU3, IAAU4, IACNF1-IACNF5, IACNF7, IACNF17, IANCM1, IANCM11, IANCM12
	Authentication		N/A				IAAU25, IANCM8, IANCM9, IANCM14
	Data Integrity		N/A				IAINT1, IAINT3, IANCM3, IANCM7, IANCM13
	Anti-replay protection		N/A				
	Bit-error Tolerance						
	Traffic Flow Security						IACNF8, IANCM2
	Dynamic Routing	N/A	N/A	N/A	N/A		
	QoS/PoS Support	N/A	N/A	N/A	N/A		IAAV1, IAAV2, IANCM4, IANCM5, IARC01-IARC03, IARC05
	Dynamic IP Addresses	N/A	N/A	N/A	N/A		
	Resource-Constrained Implementation						
	Black Media Gateway Capability				N/A		

This Table is (U//FOUO)

		Technology Category					
		FNBDT	Interop / Gateways	FNBDT Voice over IP	VoIP Call Control	IP Encryption	Required Capability (attribute from RCD)
IA Attribute	Crypto Sync Maintenance		N/A	N/A	N/A		
	Denial of Service Protection						
	Multipoint Operation					N/A	
	Key Management		N/A				IAKCM1, IAKCM3-IAKCM6, IAKCM15, IAKCM16, IAKCM18, IAKCM23, IAKCM32, IAKCM35, IAKCM38, IAKCM39, IAKCM41, IAKCM43, IAKCM45, IAKCM47, IAKCM48, IAKCM50, IAKCM53
	Clear-to-Secure Transition					N/A	
	Mobile Environment Support						
	Electronic Rekey		N/A				IAKCM44

This Table is (U//FOUO)

Table 2.3-15 reflects the gap analysis for the non-real-time application layer technologies (i.e., traditional layered application security , session security, and web services security. The mapping of RCD attributes to the IA Attributes will be provided in a future release.

Table 2.3-15: (U//FOUO) Gap Analysis for Non-real-time Application Layer Technologies

This Table is (U//FOUO)					
		Technology Categories			
		Traditional Layered Application Security	**Session Security**	**Web Services Security**	**Required Capability (attribute from RCD)**
IA Attributes	Confidentiality				
	Integrity				
	Authentication				
	Labeling				
	Access Control				
	Persistent Security				
	Standards Mature				
	Products Available				
	Technology Deployed				
This Table is (U//FOUO)					

The gaps identified in Table 2.3-16 are based upon an investigation of warfighter requirements. The assumption is that CDS technologies are to be used to meet compelling operational requirements. These requirements are categorized according to warfighter objectives, warfighter protection, and environment (security, physical, operational, etc.). The technological capabilities available to meet these requirements were categorized by interdomain transfer (i.e., guards), multiple domain access via clients and servers, and software applications (voice, collaboration, command and control, situational awareness, etc.) with multiple-domain capabilities. Supporting technologies (e.g., trusted platforms) not specifically applied to CDS will be discussed in their respective enabler descriptions.

Table 2.3-16: (U//FOUO) CDS Technology Gap Assessment

This Table is (U//FOUO)

		Technologies			
		Multiple-Domain Servers and Clients	**Guards and Controlled Interfaces**	**CDS-Aware Applications**	**RCD Capability**
Warfighter Objectives	Coordination				IAAV4, IAAV8, IACNF16
	Planning				IAAV4, IAAV8, IACNF16
	Task Dissemination				IAAV4, IAAV8, IACNF16
	Intel Assessment				IAAV4, IAAV8, IACNF16
Warfighter Protection	Indications and Warnings				IAAM8, IAAV4, IAAV8, IACNF16
	Combat ID				IAAM8, IAAV4, IAAV8, IACNF16
Warfighter Environment	Constrained Resources				IAAUD9, IAAV4, IAAV15, IAAV17
	Cognitive Workload				IACND8, IACND20, IACM11, IAIAC11, IAIAC12, IAPOL1
	Dynamic Participation				IAAM6, IAAM7, IAAM8, IACND9, IACNF15, IAIAC12, IAKCM15, IAKCM29, IAKCM33, IAKCM34, IAKCM53, IAPOL1, IARC05
	Security Environment				IAAC4, IAAC5, IAAC6, IAAM4, IAAM11, IAAM12, IAAU12, IAAUD1, IAAUD2, IAAUD3, IAAUD7, IAAUD9, IACM1, IACM5, IACM11, IACND10, IACND12, IACNF1, IACNF2, IACNF3, IACNF4, IACNF5, IACNF7, IACNF11, IACNF12, IACNF13, IACNF16, IACNF17, IAFM1, IAFM2, IAFM3, IAFM4, IAIAC3, IAIAC7, IAIL1, IAIL3, IAIL4, IAIL6, IAIL13, IAIL15, IAIL19, IAIL20, IAINT1, IAINT2, IAINT4, IAIR1, IAIR3, IAKCM29, IAKCM36 IAKCM30, IANMP5, IANRP1, IANRP2, IANRP3, IAPOL1, IAPOL3, IARC02, IARC03, IARC04, IAUAM8
	Remote Support				IAAV17, IAPOL1

This Table is (U//FOUO)

2.3.5 (U) Protection of User Information: Recommendations and Technology Timelines

(U//FOUO) The following gaps have been identified in the Protection of User Information Enabler. Without these, the benefits to be gained by this Enabler cannot be fully satisfied.

2.3.5.1 (U) Data-in-Transit

(U) The technology gaps for Data-in-Transit have been categorized as the following types—Standards, Technology, and Infrastructure.

2.3.5.1.1 (U) Standards

(U) The following gap areas have been identified in standards associated with Secure Voice applications:

- (U) Standards for providing end-to-end QoS for IP systems, specifically mechanisms for allowing QoS information to traverse red/black boundaries
- (U//FOUO) HAIPE standard updates to support dynamic routing in a multi-homed environment, QoS, dynamic black IP addresses, mobility, end-system implementations, resource-constrained implementations, and low-bandwidth high BER environments
- (U) Standards for providing interoperability between Secure Voice over IP systems and Voice over Secure IP systems
- (U//FOUO) Standards defining a common interoperable implementation of FNBDT over IP networks, including call control, gateway operation, and user media details
- (U//FOUO) Standards defining FNBDT multipoint operation (conferencing, net broadcast applications)
- (U//FOUO) Standards defining additional voice coders for FNBDT systems on specific GIG sub-networks
- (U//FOUO) Standards defining implementation and enforcement methods for applying Quality of Protection mechanisms to secure voice data
- (U//FOUO) Standards allowing Priority Service for authorized voice users

2.3.5.1.2 (U) Technology

(U//FOUO) The following gap areas have been identified in technologies associated with Secure Voice applications:

- (U//FOUO) Secure VoIP-enabled gateways
- (U//FOUO) Secure multipoint voice operation (conferencing, net broadcast applications)

(U//FOUO) Specific areas related to trusted applications requiring research include:

- (U//FOUO) Linkage between a security policy enforced by the trusted application and the security policy enforced by the host platform. This is the composition problem which has been researched off and on without satisfactory results for at least 20 years. A side issue to be examined is what happens when the trusted application is implemented on a variety of host platforms and those platforms must communicate and interoperate

- (U//FOUO) Support for complex security policies, such as dynamic access control policies like RAdAC

- (U//FOUO) Construction of self-protecting applications that can guard themselves against attacks coming through the host platform, e.g., against attacks using disk storage or input devices.

- (U//FOUO) Work is needed for all types of trusted platforms in the areas of system performance, user friendliness, and cost-effective security.

(U//FOUO) In terms of strengthening the non-repudiation service, some technical steps can be taken. As noted above, potential technical vulnerabilities include compromise of private signature keys or shared secrets; inversion of hashing algorithms; and inability to securely store and/or retrieve records. These vulnerabilities can be narrowed through use of a stronger PKI, such as a higher-assurance DoD PKI. They can be narrowed through more controls on shared secrets; and more robust storage/retrieval systems. Time-stamping and notarization systems can be made more secure against attack (e.g., through the use of trusted operating systems and/or firewalls).

(U//FOUO) Stronger proof of intent is a research area. As noted above, Alice can claim that she was not adequately presented with all the material she signed, or that the information she was presented on her screen did not match what was signed, or that her private key was used without her knowledge and consent (e.g., by a Trojan horse program operating on her computer). Defending against these claims will require much stronger computer systems. These systems must be secure against Trojan horses and other malware being present on the system. Software must be more reliable and secure to prevent modifications being made between presenting the material to Alice on her screen and it actually being signed within the system. Determining reliably that Alice was presented with the proper material, and did understand it, requires significant research breakthroughs in the area of computer-human interfaces.

2.3.5.1.3 (U) Infrastructure

(U//FOUO) The following gap areas have been identified in infrastructure associated with Secure Voice applications:

- (U//FOUO) Secure VoIP-enabled gateways

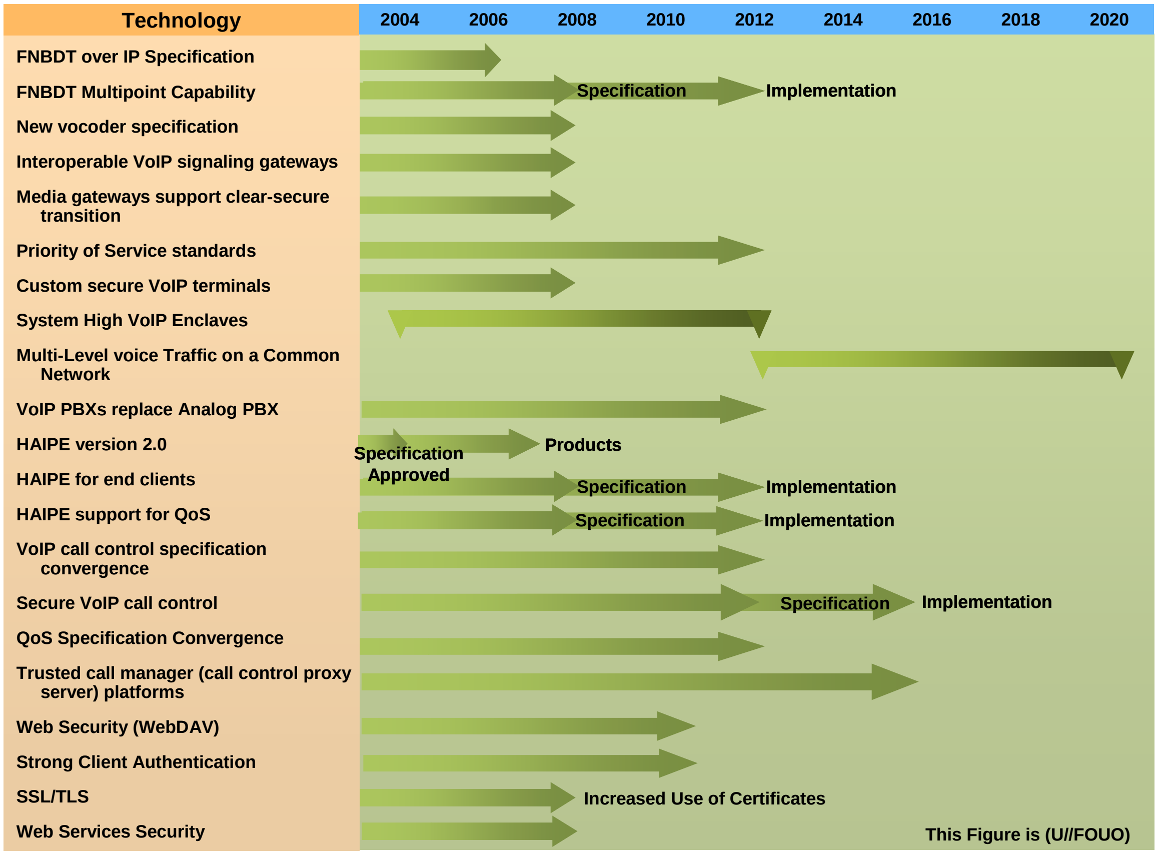


Figure 2.3-25: (U) Technology Timeline for Protection of User Information: Date in Transit

2.3.5.2 (U) Cross Domain Solutions

(U) Recommendations for CDS technologies are as follows.

- (U//FOUO) Develop trusted platforms that enable users who are not cleared to the highest level of data contained in the platform to use the platform to the level that they are cleared for.
- (U//FOUO) Develop trusted CDS workstations that allow warfighters to use applications they are accustomed to, e.g., for word processing, collaboration, situational awareness, and planning.
- (U//FOUO) Develop trusted platforms allowing multiple domain access that can function under the resource constraints of the warfighters (e.g., space, weight, and power constraints of infantry) while supporting critical functionalities (e.g., combat ID, secure voice).
- (U//FOUO) Enhance the functionality of data protection technologies to support information flows between security domains.
- (U//FOUO) Immediately developed technologies to support cross-domain real-time flows, such as voice communications and collaboration, among coalition partners

- (U//FOUO) Created standards for cross-domain technologies that focus on the reality of jointness of warfighter operations.
- (U//FOUO) Develop common CDS capabilities, adequately deploy these Joint solutions, and sufficiently train warfighters in the use of these technologies in realistic environments.

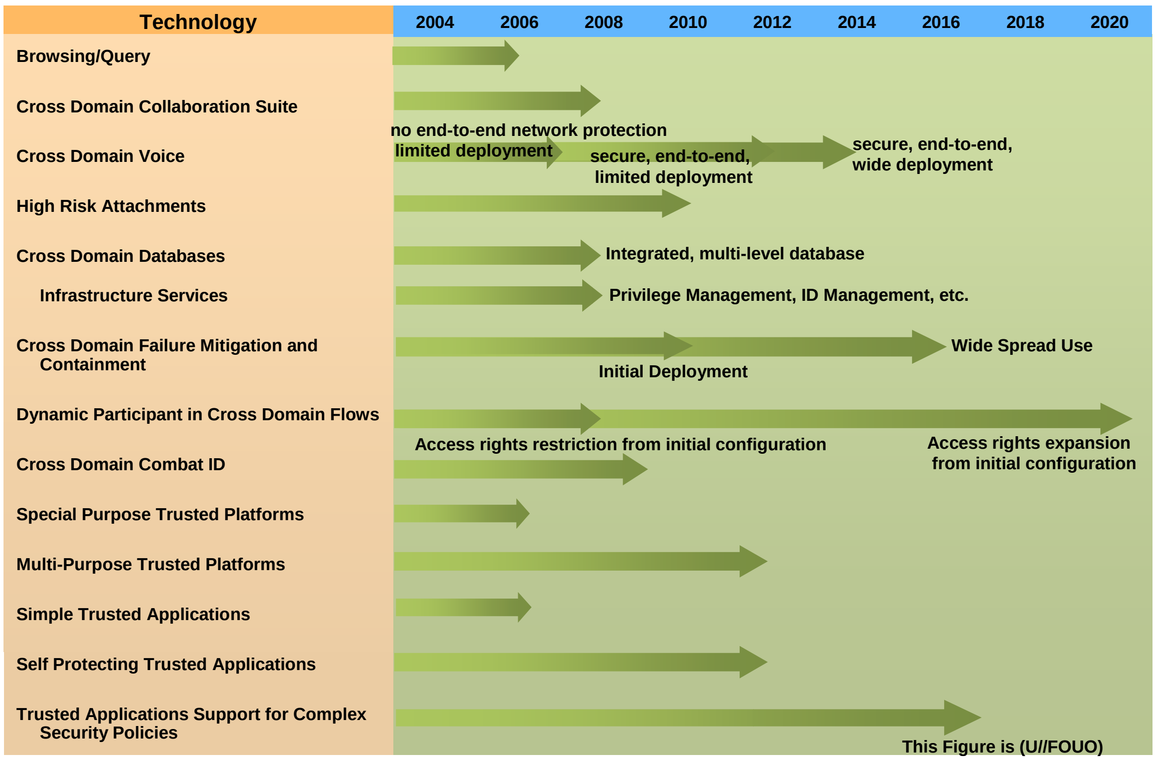


Figure 2.3-26: (U) Technology Timeline for Protection of User Information: Cross Domain Solutions

2.4 (U) DYNAMIC POLICY MANAGEMENT

(U//FOUO) Dynamic Policy Management is the establishment of digital policies for enforcing how GIG assets are managed, utilized, and protected. This includes policies for access control, Quality of Protection (QoP), Quality of Service (QoS), transport, audit, computer network defense, and policies covering the hardware and software associated with GIG assets. GIG assets include all resources within the enterprise, including physical devices (e.g., routers, servers, workstations, security components), software (e.g., services, applications, processes), firmware, bandwidth, information, and connectivity. As this list of assets shows, Policy Management is more than just information access. It also includes performance management (both transport and network management and control), enforcement of QoP, QoS, resource allocation, connectivity, and prioritization within the transport, and enforcement of access to enterprise services which are all critical to GIG availability and end-to-end data-in-transit protections.

(U//FOUO) Digital policy is the set of rules with which all actions of these assets must comply. To achieve enterprise wide (end-to-end) GIG policy management, the policies defining the rules for use of information, communications (transport), management and control functions, and service access must be integrated into a cohesive global policy. A full range of delegation of authority for policy creation and management, including intermediary policies (e.g., departmental, domain) and local (e.g., mission, COI), will still reside below the global level.

(U//FOUO) In addition, the GIG must be able to support the policies of non-GIG partners (e.g., Intelligence Community, Industry, Department of Homeland Security, State Department, Allied nations, NATO) who have GIG access. This would include establishment of an agreed approach through perhaps an assured information sharing policy for risk measurement, risk management, and risk acceptance. The policy with non-GIG entities would specify things such as U.S. access and handling rights for allied-restricted data. Reciprocally, GIG policies must address the access to GIG assets by these partners.

(U//FOUO) The dynamic aspect of policy management allows for the rapid adjustment of these rules in response to crisis situations that require either a reduction of privileges and accesses or increased latitude. These adjustments will change the criteria used to determine how resources are allocated to users and how access control decisions are made.

(U//FOUO) The GIG must not only support adjustments to policy but also conditional policies. The policy for accessing a GIG asset will specify different access rules based upon the conditions that apply to this set of information. For example, the conditions for Warfighter information may be based upon DEFCON levels and the location of the user, while the conditions for business-to-business processes may be based upon the conditions of contracting (e.g., pre-request for proposal [RFP] coordination, RFP release, contract award). Under each condition a different set of access rules would apply. A policy accounts for the various conditions that affect access to that GIG asset. The set of conditions are not expected to be global; instead policies will specify behavior for the conditions that apply.

(U//FOUO) Dynamic Policy Management allows the flexibility needed for the right data, at the right place, at the right time.

2.4.1 (U) GIG Benefits due to Dynamic Policy Management

(U//FOUO) The IA constructs used to support Dynamic Policy Management provide the following services to the GIG:

- (U//FOUO) Create and manage the set of rules that govern all GIG actions
- (U//FOUO) Provide synchronization among enterprise-wide and local policies
- (U//FOUO) Translate and distribute (push or pull) the digital policy to devices enforcing policy
- (U//FOUO) React to situational awareness conditions by changing the behavior of devices

2.4.2 (U) Dynamic Policy Management: Description

(U//FOUO) Dynamic Policy Management requires a framework to address policy management from the point of policy creation to policy installation in end devices. Included in this framework must also be the ability to dynamically update the policy in response to changing enterprise conditions. Figure 2.4-1 provides an architectural framework for discussing the functions and data flows required to perform dynamic policy management at the enterprise-level within the GIG.

(U//FOUO) Dynamic Policy Management begins with a pre-engineering phase in which the enterprise security policy is validated before entering into the enterprise. Pre-engineering of the policy is critical to ensure that policy changes do not have an adverse effect on enterprise performance or security. Typically, predictive planning through network modeling and simulation tools is used to assess the impact of candidate policy changes on operational risk, network loads, and network/application interactions and to ensure security requirements for asset usage are not violated. Local, mission-specific policies will undergo similar pre-engineering activities. Prior to deployment, these candidate policy changes should be advertised to and negotiated with the appropriate approval body. The approval body will verify that no additional issues outside of those tested in this phase are applicable to the new policy.

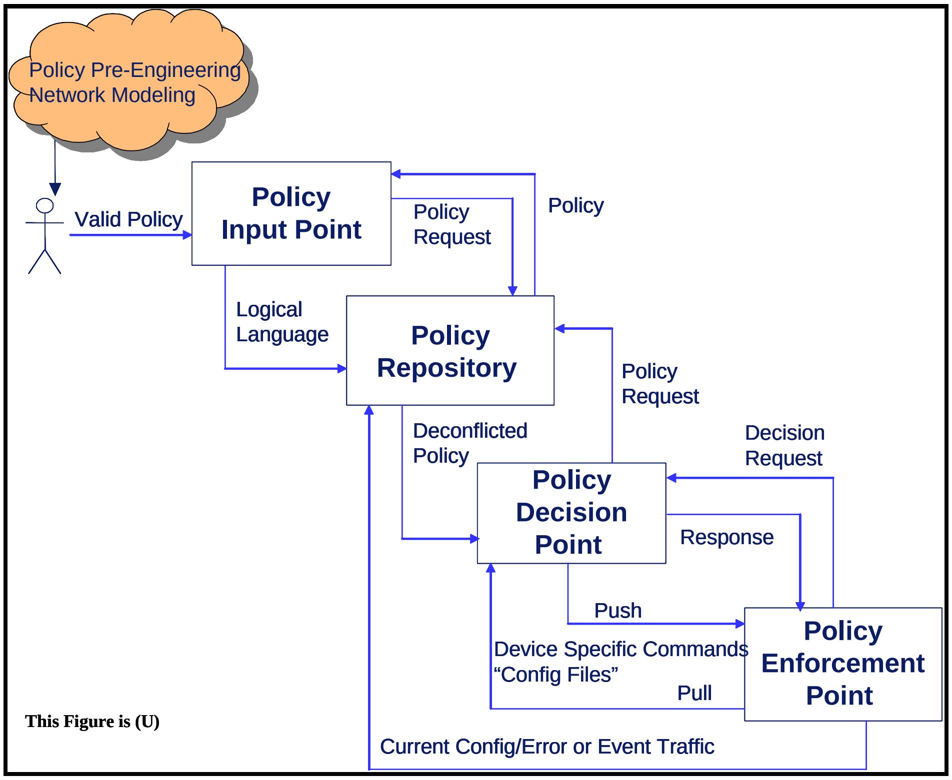


Figure 2.4-1: (U) Notional Architectural Framework for Dynamic Policy Management

(U//FOUO) Validated and approved global and local security policies enter the GIG enterprise at a policy input point. The entity entering the policy must be identified and authenticated at the input point. The input point must also determine if the entity has the proper authorizations (privileges) to enter policy. Procedures will define how an entity is granted the right to create/enter/modify policy. These privileges to enter/modify policy will be tightly controlled to ensure that false policies cannot enter the GIG enterprise. The identity of the entity that entered/modified the policy will be cryptographically bound to the policy so source and pedigree authentication can be performed. The entered policies are coded in a logical language for transfer to a policy repository. The policy is also sent to the policy repository in a human readable format so that users can read the policy and better understand its impacts.

(U//FOUO) The policy repository performs the main policy configuration management functions in the GIG. All GIG policies are securely stored at the policy repository. It also performs policy deconfliction to resolve any conflicts between the enterprise-wide policy, local, mission-specific/COI policies, and the policies of non-GIG entities (e.g., coalition partners, allies, civil, Homeland Security [HLS]) that have access to the GIG. There are specific functions performed or responses provided at a given GIG asset that may be controlled by local users and their mission-specific policies (i.e., COI policies). All other functions must be performed in accordance with the rules dictated by enterprise-wide policy. This hierarchy of policies is enforced at the policy repository.

(U//FOUO) Policy deconfliction includes the identification of policy conflicts and a resolution capability that supports automated or human adjudication between multiple policies targeted for the same device suite. These deconfliction and synchronization steps are essential to avoid vulnerabilities that could be introduced by incompatible policies. The policy repository will generate a log of detected policy conflicts and the resolution outcome so that the policy input point operator can see in English how the new deconflicted policy differs from the original.

(U//FOUO) The deconflicted policy is provided to a policy decision point (PDP). The policy decision point is a logical entity that has a centralized role in making policy decisions for itself or for other network elements that request such decisions. The PDP performs the following functions (which are further described in the following paragraphs):

- (U//FOUO) Translates policy into device specific configuration commands
- (U//FOUO) Distributes/Synchronizes policy configuration commands to affected policy enforcement points
- (U//FOUO) Services policy requests from the policy enforcement points

(U//FOUO) The PDP takes the policy rules stored in the policy repository and translates from the device-independent schema to device-specific configuration commands for the specific network devices to which the policy applies. These configuration commands program the network device to recognize the policy conditions, and when met, perform the policy action.

(U//FOUO) The PDP also services policy requests from the policy enforcement points. If a policy enforcement point does not know what to do when presented with a particular situation or set of conditions, it will make a policy request to the PDP asking for guidance. The PDP can then either make a decision or send the request further up the policy chain for resolution.

(U//FOUO) Policy distribution may take place as a result of the creation of a new policy or may be the result of a change in policy. The goal is to minimize changes in policies by defining different behaviors based upon different operational or mission conditions within a single policy. As the conditions change different behaviors are enforced. However, dynamic changes must still be supported for situations that require new behavior not anticipated in the original digital policy.

(U//FOUO) Before the policy can be pushed to the end devices, the policy's base logic must be interpreted and transformed into the specific commands understood by each targeted recipient. It is envisioned that this process be automated for the GIG. These commands must have the right level of policy enforcement granularity for the targeted recipients policy enforcement function (i.e., the policy controlling user information access may require a more dynamic and finer grained policy than a policy controlling connectivity within the Black Core).

(U//FOUO) This translation function supports the use of commercially available products such as Policy Enforcement Points (PEP). Usually, these commands take the form of configuration files. After the files are created and validated, the policy is distributed using a push or pull model. The push model would be used for policy changes that must take effect immediately because new behavior is needed under a particular condition. The pull model can be used in cases in which a policy change is scheduled to take effect at a particular time but is not critical to current operations. The targeted device pulls the updated policy from the policy decision point. Ensuring the devices receive or retrieve the updated policies in a synchronized manner is a critical aspect of policy distribution.

(U//FOUO) A PEP is a GIG asset with the responsibility of conforming to and enforcing the GIG rules (e.g., which entities can access which resources, what functions can entities perform). PEPs will be able to react and implement one or more policy rules, based on an input trigger that denotes a change in condition. These conditions will signify operational conditions or mission environment changes. Because the digital policies encode different behavior under different conditions, the PEP will implement the new rules without requiring redistribution of policy configuration information from a central source. The trigger could be automated or manual (such as an operator command). If the policy rules to implement are ambiguous (e.g., multiple conditions exist concurrently), intervention may be necessary to resolve the ambiguity.

(U//FOUO) The actual enforcement function is addressed in the Policy-Based Access Control IA System Enabler (Section 2.2). The policy management functions performed at the PEP includes policy receipt (by push or pull), policy storage, and policy error or event handling. When errors or events are detected, the PEP identifies these conditions to the policy repository for resolution. Examples of errors or events are: receipt of a configuration file that a device does not know how to use, receipt of a corrupted configuration file, or inability to pull a policy from a decision point at the specified time or under the specified condition.

(U//FOUO) Throughout the dynamic policy management architectural framework is the need for security services and mechanisms to protect the policy throughout its life cycle. From the point of creation to installation in the policy enforcement point, every GIG entity handling digital security policies must maintain the integrity of policy information for policy-at-rest and policy-in-transit throughout the management infrastructure. In addition, GIG assets must maintain integrity of the source of origin for policy throughout the management infrastructure. Confidentiality protection must be provided if the policy resident at the GIG asset requires it.

(U//FOUO) Security Services must be applied to actions within Dynamic Policy Management. Every entity sending or receiving policy information must be identified and authenticated. In addition, their privileges to send, receive, and modify policy as well as to send error or event messages to the policy repository must be validated. The integrity of the policies being promulgated must also be validated each time they are distributed and used. Other pervasive security services include the logging of all policy management transactions and the assured availability of the management infrastructure. As a critical aspect to maintain the security posture of the GIG, the availability of policy input, repository, decision, and enforcement points is vital to nearly all GIG functions.

(U//FOUO) In summary, the policy life cycle includes:

- (U//FOUO) A pre-engineering phase in which the security policy is validated before being used
- (U//FOUO) A policy creation phase, where policies enter the GIG enterprise
- (U//FOUO) Policy deconfliction to resolve the conflicts between all the policies
- (U//FOUO) Policy distribution, targeting which GIG assets should receive the digital policy and translating the base logic of the policy into device specific commands
- (U//FOUO) An installation phase in which policy is installed or replaces existing policy in end devices
- (U//FOUO) Security services and mechanisms are used to authenticate and protect the integrity, availability, and confidentiality of the policy throughout its life cycle

2.4.3 (U) Dynamic Policy Management: Technologies

(U//FOUO) The following technology areas support the Dynamic Policy Management Enabler:

- (U) Development of Policies
  - (U) Centralized vs. Distributed
  - (U) Elements of the policies
    - (U) Access Control
    - (U//FOUO) Trust Anchors
  - (U//FOUO) Policy Languages
- (U) Distribution of Policies
  - (U) Standard Protocols
  - (U) Security Issues
- (U) Policy Architectures
  - (U) Policy Directories

2.4.3.1 (U//FOUO) Development of Policies

(U) The development of policy includes the following three sub-sections:

- (U) Centralized vs. distributed
- (U) Elements of the policies
- (U) Policy languages

2.4.3.1.1 (U) Centralized vs. Distributed

2.4.3.1.1.1 (U) Technical Detail

(U) Centralized Policy Control: Several commercial products perform centralized policy management. This technology provides a centralized control of network configuration, including policy creation, maintenance, and protection. A server is used to define and store the network policies and then distribute the policies out to the remote policy enforcement points with little or no user intervention.

(U) Distributed Policy Control: Distributed policy control focuses on large dynamic networks with no central administrative control. These are independent Internet domains with dynamic topology and state information. In a multi-domain network, a number of individuals or service providers interact in a collaborative environment to provide certain services, organized according to a set of rules and policies that define how their resources can be shared among them. A distributed policy system has no centrally controlled enforcement of the policies. Consequently, there is no guarantee that policies will be followed as they are prescribed: members of a network may fail to—or choose not to—comply with the rules. If there is no way of practical (physical) enforcement of policies, then it would be useful to have a normative control mechanism for their soft enforcement (sanctions or penalties).

2.4.3.1.1.2 (U) Usage Considerations

2.4.3.1.1.2.1 (U) Implementation Issues

(U//FOUO) Current centralized policy management products are mostly product specific. For example, the Network-1 Security Solutions CyberwallPLUS product is used to configure firewalls. The McAfee® ePolicy Orchestrator® (ePO™) product is used to define policies for virus activity, desktop firewall policy, and spam and content-filtering policies. The Pedestal Software's SecurityExpressions product is used to configure Microsoft application policies.

(U//FOUO) For decentralized policy management, there are implementation issues with the synchronization of a common GIG policy amongst independent network administration systems. How do you enforce that all distributed network systems are working from the current GIG policy?

2.4.3.1.1.2.2 (U) Advantages

(U//FOUO) Centralized policy controlled systems can be configured so that local users cannot change the policy configurations at the end network devices. They can also verify current policy usage through compliance reports. This insures that the network is using the correct policy. This synchronization of policy is very important to GIG stability and overall security.

2.4.3.1.1.2.3 (U) Risks/Threats/Attacks

(U//FOUO) Centralized policy management requires strong identification, authentication, and confidentiality protection at the policy server. An attack at the centralized policy server could effect all policy enforcement points in the system.

2.4.3.1.1.3 (U) Maturity

(U) Examples of centralized policy control products includes the following:

- (U) Network-1 Security Solutions CyberwallPLUS firewall software
- (U) McAfee® ePolicy Orchestrator® (ePO™)
- (U) Pedestal Software's SecurityExpressions

(U//FOUO) The various sub-technologies of the centralized vs. distributed policy control technology area can be generally assigned Technology Readiness Level groups of Early, Emerging, or Mature.

- (U//FOUO) Centralized Policy Management—Emerging (TRLs 4- 6)
- (U//FOUO) Distributed Policy Management—Early (possibly low Emerging) (TRLs 2 – 4)

2.4.3.1.1.4 (U) Cost/Limitations

(U) When comparing centralized vs. distributed policy management, the centralized approach has less overhead cost. Performing the policy creation, verification, and distribution at a centralized site requires less personnel than a distributed approach where there could be multiple groups of people performing similar tasks.

2.4.3.1.1.5 (U) References

(U) http://www.esecurityplanet.com/prodser/article.php/1431251

(U) http://www.networkassociates.com/us/products/mcafee/mgmt_solutions/epo.htm

(U) http://infosecuritymag.techtarget.com/2002/feb/testcenter.shtml

(U) http://trantor.imit.kth.se/vinnova/DPBM.html

(U) http://www.cs.wisc.edu/condor/doc/ncoleman_tr1481.pdf

2.4.3.1.2 (U) Elements of the Policies

(U) Two technologies are discussed in this section: access control and trust anchors.

2.4.3.1.2.1 (U) Access Control

2.4.3.1.2.1.1 (U) Technical Detail

(U//FOUO) Access control policies consist of a set of rules imposed on all users and devices in the network. These rules generally rely on a comparison of the sensitivity of a resource and the possession of corresponding attributes for users or devices attempting to access the resource. These rule-based policies can be used by the GIG to enforce access control and other policies.

(U//FOUO) Some Public Key Infrastructure (PKI) programs such as Defense Message System (DMS) use rule-based policies, mostly for access control. The GIG includes policies for access control, quality of protection, quality of service, transport, audit, computer network defense, and policies covering the hardware and software associated with GIG assets. As these policies grow in complexity, so do the number of rules and the deconfliction of these rules.

(U//FOUO) These rule-based policies are first entered at the Policy Input Point (PIP) in an easily recognizable, human readable format. The PIP serves as a console for an authorized user to create new policies and edit existing policies. After the policy is created or updated, the PIP performs a translation to a base logic format that is sent to the Policy Repository. See section 2.4.2 for more details on PIP.

2.4.3.1.2.1.2 (U) Usage Considerations

2.4.3.1.2.1.2.1 (U) Implementation Issues

(U//FOUO) The GIG will have many rule-based policies. There will be enterprise-wide policies, local, mission-specific/COI policies, and the policies of non-GIG entities (e.g., coalition partners, allies, civil, HLS). Deconfliction of all these policies must take place before a policy is posted for distribution. And these rule-based policies will need to be deconflicted each time a new or update policy is introduced.

(U//FOUO) There are few deconfliction tools available today to perform this task. The KAoS policy service and Rei product have some policy confliction resolution capabilities, but these tools will need to be further developed for the GIG program. Initial versions of deconfliction tools may require operator intervention to settle conflicts between policies. As the deconfliction tools mature, this process will become more automated.

2.4.3.1.2.1.2.2 (U) Advantages

(U//FOUO) Rule-based policies can be easily expanded to define additional policy by adding new rules. This does cause more complicated attributes but with mapping each attribute category to a bit value, a very detailed user attribute can be stored in a small package.

2.4.3.1.2.1.2.3 (U) Risks/Threats/Attacks

(U//FOUO) One risk to rule-based policies is in keeping a new policy synchronized amongst the users and devices. When new policies are created with additional rules, the existing user/device attributes may not cover these rules properly and they will need to be updated. Automated distribution of policy and electronic updates of users and device attributes are required to keep rule-based policy information synchronized and working properly.

2.4.3.1.2.1.3 (U) Maturity

(U//FOUO) Basic rule-based policies are very mature in the PKI world. The DMS has been using the Security Policy Information File (SPIF) with v3 X.509 certificates for five years.

(U//FOUO) DMS uses the SPIF as a configurable access control mechanism. SPIFs contain the information needed to create and interpret security labels. Each v3 signature certificate references the SPIF, defining the security policy under which the certificate is issued. The SPIF is used to interpret Partition Rule Based Access Control (PRBAC) parameters contained in the X.509 certificate and the object security label. The SPIF is directly linked to a security policy. When a security policy is changed (i.e., the classifications or security categories are redefined), the SPIF associated with that policy must also be changed.

(U//FOUO) SPIFs are generated and signed by a root authority (i.e., trust anchor) and pushed to sub-authorities by a physical distribution path. The sub-authorities re-sign the SPIF and post the signed SPIF to the directory. The SPIF is also distributed to lower level authorities within the sub-authority's domain. Local policy dictates whether end users receive SPIFs through distribution or retrieve them from the directory.

(U//FOUO) There is enough flexibility in the SPIF to create a fairly complex implementation. Since there are no syntactic constraints on the uniqueness of displayable strings for security classifications and security categories, it is possible for independent classifications or categories to be assigned the same representation. To limit this complexity, SPIF implementers shall ensure that all human readable (displayable) or external representations of security classifications and security categories are unique within a SPIF implementation.

(U//FOUO) When two security policy domains cross-certify, there is the possibility that two or more external policy sensitivities might be mapped to a single local policy sensitivity. This many-to-one sensitivity mapping must be carefully managed to prevent unwanted changes in sensitivities when sending data across policy domain boundaries. For example, a security policy in Domain 2 may be implemented so that both Sensitivity A and Sensitivity B originating in Domain 1 will be mapped to Sensitivity X in Domain 2. The possibility of sensitivities changing when mapped between policy domains must be carefully considered when the two Security Policy Authorities develop equivalencies between their respective security policies.

(U//FOUO) The various sub-technologies of the access control technology area can be generally assigned Technology Readiness Level groups of Early, Emerging, and Mature.

- (U//FOUO) Rule based access control—Mature (TRLs 6 – 9)
- (U//FOUO) Adaptive access control—Early (TRLs 1 – 3)
- (U//FOUO) Deconfliction of policy—Early (TRLs 1 – 3).

2.4.3.1.2.1.4 (U) Standards

Table 2.4-1: (U) Access Control Standards

This Table is (U)	
Standard	**Description**
SDN.801	SDN.801 addresses concepts, tools and mechanisms for implementation of access control (AC). SDN.801 should be used to gain both a global understanding of MISSI access control, and as a guide for implementing access control features in MISSI-compliant components. SDN.801 is designed to advance from general concepts that introduce access control to more

This Table is (U)	
Standard	**Description**
	detailed information on access control tools, mechanisms, and processes as they apply to real-world communication systems.
ANSI INCITS 359-2004	This standard describes Role Based Access Control (RBAC) features that have achieved acceptance in the commercial marketplace. It includes a reference model and functional specifications for the RBAC features defined in the reference model. RBAC has become the predominant model for advanced access control because it reduces the complexity and cost of security administration in large networked applications. Many information technology vendors have incorporated RBAC into their product line, and the technology is finding applications in areas ranging from health care to defense, in addition to the mainstream commerce systems for which it was designed. The National Institute of Standards and Technology (NIST) initiated the development of the standard via the INCITS fast track process.
XACML 1.0	XACML is an XML-based language, or schema, designed specifically for creating policies and automating their use to control access to disparate devices and applications on a network.
This Table is (U)	

2.4.3.1.2.1.5 (U) Cost/Limitations

(U) GIG dynamic policy management performs policy deconfliction to resolve the conflicts between the enterprise-wide policy, local, mission-specific/COI policies, and the policies of non-GIG entities. There are limitations on how well current access control methods can support this deconfliction process.

2.4.3.1.2.1.6 (U) Alternatives

(U) XACML is an OASIS standard (Organization for the Advancement of Structured Information Standards) that describes both a policy language and an access control decision request/response language (both written in XML). The policy language is used to describe general access control requirements, and has standard extension points for defining new functions, data types, combining logic, etc. The request/response language lets you form a query to ask whether or not a given action should be allowed, and then interpret the result. This resulting response always includes an answer about whether the request should be allowed, using one of four values: Permit, Deny, Indeterminate (an error occurred or some required value was missing, so a decision cannot be made) or Not Applicable (the request can't be answered by this service).

2.4.3.1.2.1.7 (U) References

(U) FORTEZZA® Security Management Infrastructure (SMI) Concept of Operation CONOP) for CipherNET® 3000 CAW 5.0

(U) SDN.801: ACCESS CONTROL CONCEPT AND MECHANISMS

(U) http://www.oasis-open.org/committees/download.php/2713/ Brief_Introduction_to_XACML.html

2.4.3.1.2.2 (U//FOUO) Trust Anchors

2.4.3.1.2.2.1 (U) Technical Detail

(U//FOUO) The purpose of a trust anchor is to serve as a baseline for the validation of some entity/action. The trust anchor is something that has been accepted through out-of-band means as being valid and reliable. For example, it can be a public key or certificate corresponding to a private key. Without this baseline, there is no sound way of validating anything else.

(U//FOUO) In some systems, the trusted anchor is called a trusted root or root authority. The trusted root or root authority is the point at which trust begins in a PKI system. The root authority is the certification authority that certifies the existence and quality of other certification authorities in the particular PKI that you wish to use. The business and Internet communities are not waiting for some over-arching system to be put into place by governments or agencies. They are seizing opportunities as they arise—putting in place systems that they trust and selecting their own root authorities.

(U//FOUO) The initial loading of a trust anchor in the system MUST be by a trusted out of band means. If you receive a trust anchor over the network—how do you know it's good? You have no trust anchor to use to validate the new one, so you either take a chance that you're being spoofed and accept it (and open yourself up to lots of attacks), or you refuse to accept it because you can't validate it. That is why it is so important that the initial loading of a trust anchor comes from a highly trusted source.

(U//FOUO) With respect to dynamic policy management, how does the policy input point know to trust the person requesting to create or edit GIG policies? How does the policy enforcement point verify the policy configuration file received from the policy decision point? The answer to these questions starts with the trust anchor.

(U//FOUO) Once the initial loading of a trust anchor has been accomplished, it can be updated or transferred securely over the net. See RFC 3157 for details of the Securely Available Credentials (SACRED) protocol, which can be used to securely transfer credentials.

(U//FOUO) The trust anchor and the personnel managing the trust anchor are the heart of the trust in PKI and other authentication-based systems. The consequences of compromise to a trust anchor by malicious intent, inadvertent errors, or system failures can be severe. Hence, this trust anchor must be diligently protected. Such protection can be provided by placing all cryptographic key management and encryption/decryption functions into a trusted/tamper-proof hardware device rather than residing in software on a host computer.

(U//FOUO) Trust anchors operate under a set of rules or policies that describe both the physical and electronic protection of the trust anchor information. Failing to follow these rules and policies could cause the revocation or compromise of the trust anchor, affecting all authorities, users, and devices whose authentication path is based on that trust anchor.

2.4.3.1.2.2.2 (U) Usage Considerations

2.4.3.1.2.2.2.1 (U) Implementation Issues

(U//FOUO) The main implementation issue with trust anchors is the initial delivery of the trust anchor information. If you receive a trust anchor over the network—how do you know it's good? It is very important that the initial loading of a trust anchor comes from a highly trusted source.

2.4.3.1.2.2.2.2 (U) Advantages

(U//FOUO) Trust anchors provide a fairly simple and straightforward method of verifying authentication paths for users, devices, and organizations. With the help of compromise lists and revocation lists, the trust anchor provides the information needed to determine if a message or data is from a valid source.

2.4.3.1.2.2.2.3 (U) Risks/Threats/Attacks

(U//FOUO) Trust anchors must be protected from both physical and electronic attacks due to the implications of a revocation or compromise. Trust anchors should be stored in well-protected locked areas. Multi-person access to the physical location will reduce the risk of attacks. Multi-person access to the workstation or system containing the trust anchor would further reduce attacks. Personnel operating the trust anchors should be highly trusted individuals.

2.4.3.1.2.2.3 (U) Maturity

(U//FOUO) PKI systems have been using trust anchors for over ten years. The trust anchor in a PKI system is usually called the root authority. Some PKI systems also support cross certificates, which allow certificate path validation between users under different trust anchors.

(U//FOUO) The various sub-technologies of the trust anchor technology area can be generally assigned Technology Readiness Level groups of Early, Emerging, and Mature.

- (U//FOUO) PKI root authority—Mature (TRLs 6 – 9)
- (U//FOUO) Cross registration between trust anchors—Emerging (TRLs 4 – 6)
- (U//FOUO) Trust anchor initial load and updates—Emerging (TRLs 4 – 6).

2.4.3.1.2.2.4 (U) Standards

Table 2.4-2: (U) Trust Anchor Standards

This Table is (U)

Standard	Description
RFC 3157	This document identifies a set of requirements for credential mobility. Using SACRED protocols, users will be able to securely move their credentials between different locations, different Internet devices, and different storage media as needed.

This Table is (U)

2.4.3.1.2.2.5 (U) References

(U) FORTEZZA® Security Management Infrastructure (SMI) Concept of Operation CONOP) for CipherNET® 3000 CAW 5.0

2.4.3.1.3 (U) Policy Languages

2.4.3.1.3.1 (U) Technical Detail

(U//FOUO) Policy Languages are used to define policy statements that can be used by networking hardware such as routers, firewalls, and guards. These policy statements can be used for routing, access control, and QoS purposes.

(U//FOUO) Several policy languages exist which may be appropriate for application in the GIG: Routing Policy Specification Language, Path-based Policy Language, Security Policy Specification Language, KeyNote, and Extensible Access Control Markup Language (XACML). But most of these languages were designed for one thing, such as generate routing tables, QoS using differentiated service code points, access control using access control lists (ACLs), etc.

(U//FOUO) GIG requires dynamic policy management that handles all the required GIG policies, including: access control, RAdAC, QoP, QoS, transport, audit, computer network defense, and policies covering the hardware and software associated with GIG assets. To do that, either multiple policy languages will be needed to create the overall GIG policy or a more robust policy language needs to be developed that will support all the GIG policies. Some existing policy languages such as Ponder, KAoS, Rei, and XACML are flexible in that they allow you to define new policy within the language. GIG should further research these flexible policy languages to see which would be best suited for the GIG policies.

2.4.3.1.3.2 (U) Usage Considerations

(U//FOUO) RAdAC will need specific capabilities in its access control policy but should fold into the larger GIG dynamic policy effort. Some potential technologies that could support access control policy include WS-Policy, Standard Deontic Logic such as that implemented in Rei or Ponder, and artificial intelligence constructs in PROLOG, decision trees, or fuzzy logic. This section assumes that the distributed functionality (e.g., secure update, revocation, currency validation, and caching for off-line use) is provided by the dynamic policy enabler and thus focuses only on RAdAC-specific digital policy needs.

(U//FOUO) Dynamic Access Control Policy serves as an input to the RAdAC model in order to control its behavior. In this usage, the policy must be expressive enough to dictate some or all of the following access control characteristics:

- (U//FOUO) Minimum number of required inputs to calculate risk and operational need
- (U//FOUO) Relative weighting of the various inputs for risk and operational need
- (U//FOUO) Relative weighting of risk versus operational need for the final decision
- (U//FOUO) Ability to understand (in human readable terms) the limiting factors (LIMFAC) that contributed to a failed access attempt

- (U//FOUO) Ability to express stateful access control rules (e.g., successive failed access attempts)
- (U//FOUO) Ability to express policy according to enterprise and COI roles
- (U//FOUO) Ability to negotiate two or more conflicting access control rules
- (U//FOUO) Ability to negotiate access control policy with neighboring security domains in order to define an access control boundary interface that is agreeable to both sides
- (U//FOUO) Ability to express and automatically select between multiple policies based on nationality or security domain
- (U//FOUO) Ability to express more granular or more restrictive access control policies at each successive echelon down the chain of command
- (U//FOUO) Ability to dynamically tighten or loosen access control policy based on situation (INFOCON, proximity to enemy forces, etc.).
- (U//FOUO) Ability to do all of this very quickly so as not to become the system bottleneck

(U//FOUO) In this first role influencing RAdAC behavior, the policy must somehow be able to handle policy exceptions (termed “dispensations” in some deontic languages) that are able to authorize otherwise disallowed actions—but only for a limited time period and only for a well-defined set of actions.

(U//FOUO) Due to national law or immutable operational policy, care has to be taken to constrain where dispensations themselves are allowed and not allowed within the policy language. For example, dispensations may be allowed for dissemination of a classified document to a cleared User without formal access approval, given compelling operational need but may never be allowed for an uncleared User. Dispensations may be the most appropriate way for digital policy to annotate and reason about a commander’s or supervisor’s consent for a User’s operational need to know a particular piece of information.

(U//FOUO) Dynamic Access Control Policy also requires expressiveness for RAdAC output. For instance, the policy engine may recognize a specific request as having a compelling operational need but having too risky an IT Component to release the information to. In this case, policy should be expressive enough to conclude that an alternate path (alternate Course Of Action, or COA) for this LIMFAC should be examined before arriving at a final access decision. In this role, policy must be expressive enough to dictate the following alternate COA determinations:

- (U//FOUO) Alternate enterprise routing evaluation to obtain higher QoP from end to end
- (U//FOUO) Digital rights restrictions to limit the risk of disclosure or further dissemination
- (U//FOUO) Automatic sanitization through a guard (or originator) process prior to release

- (U//FOUO) Evaluation of nearby neighbors or superiors who might have more robust IT Components for handling the data as-is

(U//FOUO) In this second role influencing RAdAC output, the policy must be tightly integrated with the policies that affect management of the IT Components. This avoids situations where RAdAC allows access through a given enterprise route but then the enterprise routes the information over a different path because of other decision metrics. Digital rights policy enforcement must be tightly integrated with the end user equipment portion of IT Components so that the rights embedded with the information object are strictly enforced.

(U//FOUO) Finally, the policy must be robust enough to meet extremely stringent false negative and false positive rates. Since RAdAC would be replacing the traditional Mandatory Access Control model objectively, false positives in particular cannot be tolerated for risk of information disclosure. Dispensations for exception handling must be constrained in such a way that guarantees select portions of digital access control policy will comply with national law.

2.4.3.1.3.2.1 (U) Implementation Issues

(U//FOUO) Current policy management products are mostly vendor specific. There are many forms of policy languages for covering routing, QoS, or access control.

- (U) Routing Policy Specification Language (RPSL) was developed by the IETF Routing Policy System Working Group (RFC 2622 and RFC 2725). RPSL allows a network operator to specify routing policies at various levels in the Internet hierarchy; for example at the autonomous system level. At the same time, policies can be specified with sufficient detail in RPSL so that low-level router configurations can be generated from them. RPSL is extensible and new routing protocols and new protocol features can be introduced at any time.
  - (U) Tier 1 ISP in Australia designed and built Connect's RPSL-based system to manage routing policy and configure routers. Problem: Policy can easily get very complex and result in very complex router configuration.
- (U) Ponder Policy Specification Language: Ponder is a declarative, object-oriented language for specifying management and security policies for distributed systems. It is a role-based access control. Ponder is a product of the Imperial College of Science, Technology, and Medicine in London, England. It has been developed as part of ongoing research being carried out by the group into the use of policy in distributed systems management. Ponder is a general-purpose management language for specifying what actions are performed and how to allocate resources when specific events occur.
  - (U) The Ponder toolkit includes the following:
    - (U) Ponder Compiler: A Compiler framework for the Ponder policy specification language. It supports the main features of the Ponder grammar. It consists of a Syntax Analyzer, a two-pass Semantic Analyzer, and the default Java Code Generator for Obligation and Refrain Policies, and XML code generator.

    - (U) Ponder Policy Editor: A customizable text editor for the Ponder language, written in Java. It has all the basic features of a text editor and includes features that make text editing Ponder Policies easy.
    - (U) Ponder Management Toolkit: A Management Toolkit Framework, designed to allow for the addition of tools to be managed from a central management console.

  - (U) Ponder also has built-in tools for performing both runtime checking of policy rules and offline checking of policy rules.

- (U) The Security Assertion Markup Language (SAML) is a planned standard for interoperability among Web services security products. SAML is developed and maintained by the Organization for the Advancement of Structures Information Standards (OASIS) organization's XML-Based Security Services Technical Committee (SSTC). SAML defines a common XML framework for exchanging security assertions between entities for the purpose of exchanging authentication and authorization information.

- (U) Extensible Access Control markup Language (XACML). XACML is an OASIS standard that describes both a policy language and an access control decision request/response language (both encoded in XML). The policy language is used to describe general access control requirements. It has standard extension points for defining new functions, data types, combining logic, etc. The request/response language lets you form a query to ask whether or not a given action should be allowed and then interpret the result.

  - (U) Parthenon Software has produced a suite of Policy products based on XACML. It identifies an XML-based language that is used to describe access control requirements for online resources. The intent is to allow for efficient machine parsing of arbitrarily complex security policies.

  - (U) Sun's XACML was developed in Sun Microsystems Laboratories, part of Sun Microsystems, Inc., as an open source implementation of the OASIS XACML standard, and was written in the Java™ programming language. This product provides complete support for all the mandatory features of XACML as well as a number of optional features. Specifically, there is full support for parsing both policy and request/response documents, determining applicability of policies, and evaluating requests against policies. All of the standard attribute types, functions, and combining algorithms are supported, and there are interfaces for adding new functionality as needed. Sun is looking at adding features to connect XACML and things like SAML or Lightweight Directory Access Protocol (LDAP), and strong tools support.

  - (U) Lagash Systems XACML.NET is an implementation of the XACML specification released by OASIS in purely .Net code (C#) that can be used by anyone in the .Net developer community. XACML.NET is under the Mozilla public license (MPL) 1.1 so any software under a license compatible with MPL can use this code.

- (U) KeyNote is a flexible trust-management system designed to work well for a variety of large- and small-scale Internet-based applications. KeyNote was designed and developed in 1997 by representatives from AT&T Labs, Yale University, and the University of

Pennsylvania. It provides a single, unified language for both local policies and credentials. KeyNote policies and credentials, called assertions, contain predicates that describe the trusted actions permitted by the holders of specific public keys. KeyNote assertions are essentially small, highly structured programs. A signed assertion, which can be sent over an untrusted network, is also called a credential assertion. Credential assertions, which also serve the role of certificates, have the same syntax as policy assertions but are also signed by the principal delegating the trust.

  - (U) KeyNote is described in RFC-2704. It has no restrictions on its use and distribution. The KeyNote Toolkit is a C language open-source reference implementation and can be obtained at http://www.crypto.com/trustmgt/kn.html

- (U) Rei was developed by the eBiquity Group, a research organization that consists of faculty and students from the Department of Computer Science and Electrical Engineering (CSEE) of UMBC. Rei is a policy language based on OWL-Lite (Web Ontology Language with a restricted vocabulary) that allows policies to be specified as constraints over allowable and obligated actions on resources in the environment. Rei also includes logic-like variables, which give it the flexibility to specify relations like role value maps that are not directly possible in OWL. Rei includes meta policy specifications for conflict resolution, speech acts for remote policy management, and policy analysis specifications like what-if analysis and use-case management—making it a suitable candidate for adaptable security in the environments under consideration. The Rei engine, developed in XSB (extended Prolog), reasons over Rei policies and domain knowledge in Resource Description Framework (RDF) and OWL to provide answers about the current permissions and obligations of an entity, which are used to guide the entity's behavior.

- (U) The Web Services Policy Framework (WS-Policy) was developed by BEA Systems Inc., IBM Corporation, Microsoft Corporation, and SAP AG. The WS-Policy specification provides a general-purpose model and corresponding syntax to describe and communicate the policies of a Web service. The goal of WS-Policy is to provide the mechanisms needed to enable Web Services applications to specify policy information. WS-Policy by itself does not provide a negotiation solution for Web Services. WS-Policy is a building block that is used in conjunction with other Web Service and application-specific protocols to accommodate a wide variety of policy exchange models.

- (U) Knowledgeable Agent-oriented System (KAoS) is a collection of component agent services compatible with several popular agent frameworks, including Nomads, the DARPA CoABS Grid, the DARPA ALP/Ultra*Log Cougaar framework, CORBA, and Voyager. The adaptability of KAoS is due in large part to its pluggable infrastructure based on Sun's Java Agent Services (JAS). KAoS policy services is developed by The Institute for the Interdisciplinary Study of Human & Machine Cognition (IHMC) under NASA and DARPA sponsorship.

  - (U) KAoS policy services allow for the specification, management, conflict resolution, and enforcement of policies within domains. Policies are represented in DAML (DARPA Agent Markup Language) as ontologies. The KAoS Policy Ontologies (KPO) distinguish between authorizations (i.e., constraints that permit or forbid some action) and obligations (i.e., constraints that require some action to be performed—or else serve to waive such a requirement). Through various property restrictions in the action type, a given policy can be variously scoped, for example, either to individual agents, to agents of a given class, to agents belonging to a particular group, or to agents running in a given physical place or computational environment (e.g., host, VM).

  - (U) KAoS framework supports dynamic runtime policy changes and is extensible to a variety of execution platforms that might be simultaneously running with different enforcement mechanisms. Currently KAoS supports agent platforms implemented in Java and Aroma, but could be adapted to work with other platforms for which policy enforcement mechanisms are written.

- (U) Semantic Web Rule Language (SWRL) was produced as part of the DARPA DAML Program. SWRL is built on top of the W3C Ontology layer (OWL DL and OWL lite and a subset of RuleML, a Rule Markup Language). As such SWRL implements Frame Logic that unfortunately omits the Deontic Modal Operators, (i.e., ‘P’ "it is permitted that", 'O' "it is obligatory that", and 'F' "it is forbidden that"). SWRL can be used as the logic layer in Berners-Lee’s seven-layer model of the Semantic Web. See Figure 2.4-2 below.

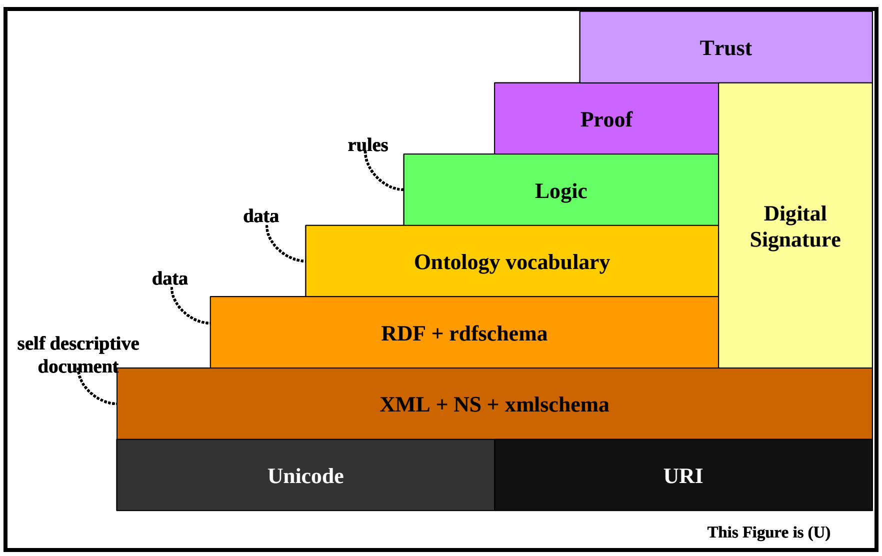


Figure 2.4-2: (U) Berners-Lee’s Seven Layer Model of the Semantic Web

2.4.3.1.3.2.2 (U) Advantages

(U//FOUO) Having one policy language would make it easier for the person managing the GIG policy to understand. A single common policy language would also greatly simplify the GIG policy management components (i.e., Policy Input Point, Policy Repository, and Policy Decision Points).

(U//FOUO) A single policy language would also simplify the translation to device specific configuration files needed at the policy enforcement points.

2.4.3.1.3.2.3 (U) Risks/Threats/Attacks

(U//FOUO) Need to verify that what is put in the language actually gets translated into the device configuration files correctly. This will require verification testing prior to a new policy entering the GIG.

(U//FOUO) Also need authentication and integrity protection on the messages to prevent spoofing and possibly confidentiality to protect sensitive policy data. This can be either implemented directly in the policy protocol—or implemented in a lower layer protocol, like IPsec or transport layer security (TLS).

2.4.3.1.3.3 (U) Maturity

(U//FOUO) Several policy languages are being used by commercial products today:

- (U) Sun's xacml: http://sunxacml.sourceforge.net/
- (U) Ponder toolkit: http://www-dse.doc.ic.ac.uk/Research/policies/ponder.shtml
- (U) KeyNote toolkit: http://lists.netfilter.org/pipermail/netfilter/1999-October/002634.html
- (U) KAoS toolkit: http://www.ihmc.us/research/projects/KAoS/
- (U) Cisco's QoS Policy Management: http://www.cisco.com/warp/public/cc/pd/wr2k/qoppmn/
- (U) Nortel's Optivity Suite: http://www.nortelnetworks.com/products/01/optivity/policy/index.html

(U//FOUO) The various sub-technologies of the policy language technology area can be generally assigned Technology Readiness Level groups of Early, Emerging, and Mature.

- (U//FOUO) Routing and access control languages—Mature (TRLs 7 –9)
- (U//FOUO) Extensible policy languages—Emerging (TRLs 4 – 6)
- (U//FOUO) Security incorporated into policy languages—Early (TRLs 1 – 3)
- (U//FOUO) Verification/test of policy languages—Early (TRLs 1 – 3)
- (U//FOUO) Handling policy conflicts—Early (TRLs 1 – 3).

2.4.3.1.3.4 (U) Standards

Table 2.4-3: (U) Policy Language Standards

This Table is (U)

Standard	Description
Extensible Access Control markup Language	XACML provides fine-grained control of authorized activities, the effect of characteristics of the access requestor, the protocol over which the request is made, authorization based on classes of activities, and content introspection.

This Table is (U)

Standard	Description
(XACML)	
Routing Policy Specification Language (RPSL)	RPSL allows a network operator to be able to specify routing policies at various levels in the Internet hierarchy. Policies can be specified with sufficient detail in RPSL so that low-level router configurations can be generated from them. RPSL is extensible; new routing protocols and new protocol features can be introduced at any time.
Rei	A declarative policy language for describing policies over actions. It is possible to write Rei policies over ontologies in other semantic web languages.
KeyNote	KeyNote provides a simple language for describing and implementing security policies, trust relationships, and digitally signed credentials.
SDN.801	SDN.801 provides guidance for implementing access control concepts using both public key certificates and attribute certificates.
Security Assertion Markup Language (SAML)	SAML is an XML framework for exchanging authentication and authorization information.
Ponder	Ponder is a language for specifying management and security policies for distributed systems.
KAoS	KAoS policy services allow for the specification, management, conflict resolution, and enforcement of policies within domains.

This Table is (U)

2.4.3.1.3.5 (U) Cost/Limitations

(U//FOUO) The policy language used by GIG will need to cover all GIG policies. This includes policies for access control, QoP, QoS, transport, audit, computer network defense, and policies covering the hardware and software associated with GIG assets.

2.4.3.1.3.6 (U) Dependencies

(U//FOUO) Need compilers to translate the policy language into configuration files that are used by the policy enforcement points. These configuration files are mostly vendor specific so a compiler would need to output many different formats.

(U//FOUO) Also need testing and verification tools to test the policy language statements prior to distribution to the operational environment.

2.4.3.1.3.7 (U) Alternatives

(U) Generate new policy language to securely cover all the GIG policy management needs. This would be an expensive and time-consuming task.

2.4.3.1.3.8 (U) References

(U) http://www.parlay.org/about/policy_management/index.asp

(U) www-106.ibm.com/developerworks/library/ws-secpol/

(U) http://sunxacml.sourceforge.net/guide.html

(U) RFC 2622

(U) http://www.comsoc.org/ni/private/2001/jan/stone.html

(U) http://www.doc.ic.ac.uk/~mss/Papers/Ponder-Policy01V5.pdf

(U) http://www.cis.upenn.edu/~keynote/

(U) http://rei.umbc.edu/

(U) http://www.ihmc.us/research/projects/KAoS/FinalIHMC_DEIS.pdf

(U) http://www.parthenoncomputing.com

(U) http://sunxacml.sourceforge.net/

(U) http://mvpos.sourceforge.net/

(U) http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnglobspec/html/ws-policy.asp

(U) http://www.wiwiss.fu-berlin.de/suhl/bizer/SWTSGuide/KAoS/KAoS_Policy_03.pdf

(U) http://www.ihmc.us/research/projects/KAoS/

(U) http://www.daml.org/2003/11/swrl/rules-all.html

2.4.3.2 (U) Distribution of Policies

2.4.3.2.1 (U) Standard Protocols

2.4.3.2.1.1 (U) Technical Detail

(U//FOUO) Distribution of dynamic material is required to configure the policy enforcement points, through the use of GIG policy files. After the files are created and validated, the policy is distributed using a push or pull model. The push model would be used for policy changes that must take effect immediately because new behavior is needed in reaction to the current condition. The pull model can be used in cases in which a policy change is scheduled to take effect at a particular time and is not critical to current operations.

(U//FOUO) Policy distribution extends from the policy input point to the Policy Enforcement Points (PEP). PEPs are those GIG assets that enforce the GIG rules. (See section 2.4.2 for more information on PEP.) PEPs include routers, firewalls, guards, and other networking equipment that require configuration files to enforce policy. Most PEP equipment is currently configured manually by network support personnel. But some policy management products are using directories to store policy configuration information and Light-weight Directory Access Protocol (LDAP) to distribute the configuration files.

(U//FOUO) These policy enforcement configuration files are generally vendor specific and only support routing and access control policy decisions. The policy distribution point will need to know the type of PEP when distributing new policy so that the policy can be in the correct configuration format for the specific PEP.

(U//FOUO) It is highly critical that the GIG program work with PEP vendors to expand PEP capabilities and possibly standardize policy enforcement configuration files to reduce policy management overhead. Common Open Policy Service (COPS) protocol and Command Line Interface (CLI) commands are two enforcement configuration formats currently being used.

(U//FOUO) COPS is a query and response protocol that the PDP and PEP can use to exchange policy information. COPS uses the Transmission Control Protocol (TCP) to transfer the messages.

(U//FOUO) There are other options for distributing the policy updates. Administrators can send users an email with a URL where users can download the update, or use a facility such as Microsoft's System Management Server (SMS) to automatically push the updates out to distributed end points.

(U//FOUO) Another alternative is to use the CyberwallPLUS policy pull feature. Each time a user logs on to the network, the software checks a central policy database to ensure the user has the most current policy configuration.

2.4.3.2.1.2 (U) Usage Considerations

2.4.3.2.1.2.1 (U) Implementation Issues

(U) Current policy management products are mostly vendor specific. Policy distribution formats must be agreeable with the network products receiving the policy information.

2.4.3.2.1.2.2 (U) Advantages

(U//FOUO) Automating the distribution of policy information would be a significant savings over the current manual configuration of PEPs. To fully take advantage of this automated distribution, the integrity and authentication of the delivery must be verifiable to insure that the policy was received unchanged from a trusted source.

(U) Having a common distribution protocol would greatly simplify the distribution process to the network components.

2.4.3.2.1.2.3 (U) Risks/Threats/Attacks

(U//FOUO) Policy data must be protected from the time the policy is created at the policy input point to the time the policy reaches the policy enforcement points. This requires identification and authentication of the person creating new policies. It also requires authentication, integrity, and confidentiality of the policy data as it passes through the GIG policy management system.

2.4.3.2.1.3 (U) Maturity

2.4.3.2.1.3.1 (U) DMS Example

(U//FOUO) DMS has a trusted policy distribution system with both manual and automated procedures. With DMS, the rule-based access control policy is held in the SPIF. An External Source, such as a policy making body, generates the security policies used by DMS. This information is delivered to a root authority in an unsigned SPIF format on a trusted physical path. The root authority reviews and approves the security policy before signing the SPIF. After signing an SPIF, the root authority distributes it to the subordinate authorities that support the security policy defined in the SPIF. The root authority can maintain multiple SPIFs, but the subordinate authorities only need to receive the SPIFs for the security policy(s) they support.

(U//FOUO) The sub-authority verifies the received SPIF has been signed by the root authority and is valid. Next, the sub-authority removes the root authority signature, updates the issuer and date information, and re-signs the SPIF. The sub-authority then posts the SPIF to the directory and distributes the SPIF to the rest of the authority hierarchy.

(U//FOUO) User applications and devices using the SPIF will periodically retrieve the SPIF from the directory, verify the signature of the SPIF, and use the SPIF for access control decisions.

2.4.3.2.1.3.2 (U) Vendor Distribution Example

(U//FOUO) Most network component vendors (e.g., Cisco, Juniper, Ciena, and Nortel) have configuration formats and distribution methods that are specific to their equipment. Distribution methods include LDAP, File Transfer Protocol (FTP), Telnet, and Secure Server Protocol (SSP).

(U//FOUO) The various sub-technologies of the policy distribution technology area can be generally assigned Technology Readiness Level as follows.

- (U//FOUO) Distribution protocols—Mature (TRLs 7 – 9)
- (U//FOUO) PEP configuration file standard—Early (TRLs 1 – 3)

2.4.3.2.1.4 (U) Standards

Table 2.4-4: (U) Distribution Standards

This Table is (U)	
Standard	**Description**
LDAP	LDAP is an Internet protocol used to look up information from a LDAP server or directory. LDAP servers index all the data in their entries, and "filters" may be used to select just the information you want. "Permissions" and “authentications” can be set by the administrator to allow only certain people to access the LDAP database, and optionally keep certain data private. Reference http://www.ldap-directory.org/rfc-ldap for a list of LDAP RFCs.
File Transfer Protocol (FTP)	File Transfer Protocol (FTP), a standard Internet protocol, is the simplest way to exchange files between computers on the Internet. FTP is an application protocol that uses the Internet's TCP/IP protocols.

This Table is (U)

Standard	Description
	Reference RFC959: http://www.w3.org/Protocols/rfc959/
Common Open Policy Service (COPS)	The Common Open Policy Service (COPS) protocol is a simple query and response protocol that can be used to exchange policy information between a policy server (PDP) and its clients (PEPs). Reference http://www.networksorcery.com/enp/protocol/cops.htm for a list of COPS related RFCs
Microsoft's SMS	SMS provides a solution for change and configuration management for the Microsoft platform, enabling organizations to provide relevant software and updates to users quickly and cost effectively.
Telnet	The Telnet program allows you to connect your PC to a server on the network using a username and password. You can then enter commands through the Telnet program, and they will be executed as if you were entering them directly on the server console.

This Table is (U)

2.4.3.2.1.5 (U) Dependencies

(U//FOUO) PEP configuration formats are mostly vendor specific. Creating a standard for this configuration format would require support from many network component vendors.

2.4.3.2.1.6 (U) Alternatives

(U//FOUO) For policy distribution, there are many existing protocols that can be used to safely distribute the GIG policy throughout the system.

(U//FOUO) GIG-developed common protocol for format of all GIG policy enforcement points.

2.4.3.2.1.7 (U) Complementary Techniques

(U//FOUO) Security features can also be applied to policy distribution if required by the GIG program. Directories can be configured to limit write access to the policy information so only authorized persons can create and update GIG policy information stored in the directory.

(U//FOUO) Authentication and confidentiality can also be applied to the policy distribution by adding additional levels of protection to the policy data. A protocol such as Secure Sockets Layer (SSL) allows the server and client to authenticate each other and to negotiate an encryption algorithm and cryptographic keys before the application protocol transmits or receives its first byte of data. One advantage of SSL is that it is application-protocol independent.

2.4.3.2.1.8 (U) References

(U) FORTEZZA® Security Management Infrastructure (SMI) Concept of Operation CONOP) for CipherNET® 3000 CAW 5.0

(U) http://wp.netscape.com/eng/ssl3/ssl-toc.html

(U) http://www.nortelnetworks.com/products/01/optivity/policy/index.html

(U) http://www.parlay.org/about/policy_management/index.asp

2.4.3.2.2 (U) Security Issues

2.4.3.2.2.1 (U) Technical Detail

(U//FOUO) Policy data must be protected from the time the policy is created at the policy input point to the time the policy reaches the policy enforcement points. This requires identification and authentication of the person creating new policies. It also requires authentication and integrity of the policy data as it passes through the GIG policy management system.

(U//FOUO) Policy data can provide great value to an attacker to know exactly what rules the infrastructure is enforcing. Confidentiality may also be required if the policy data contains sensitive data. Having a common configuration file format would also make it easier for an attacker to understand policy changes when they are sent to the PEPs. This is another reason confidentiality should be applied to this enforcement configuration file so outside sources cannot change or see the PEP's configuration.

(U//FOUO) Policy repository directories can be configured to limit the read and write access to policy information so only authorized persons can read and update GIG policy information stored in the directory.

(U//FOUO) Authentication and confidentiality can also be applied to the policy distribution by adding more levels of protection to the policy data. A protocol such as SSL allows the server and client to authenticate each other and to negotiate an encryption algorithm and cryptographic keys before the application protocol transmits or receives its first byte of data. One advantage of SSL is that it is application-protocol independent.

2.4.3.2.2.2 (U) Usage Considerations

2.4.3.2.2.2.1 (U) Implementation Issues

(U//FOUO) Currently, none of the policy languages incorporate the security features required for secure GIG dynamic policy distribution. So either a new GIG-defined protocol could be developed that includes the security features or existing security protocols (e.g., SSL, IPsec, or TLS) can be added to the policy distribution procedures.

2.4.3.2.2.2.2 (U) Advantages

(U//FOUO) Using a COTS solution for policy distribution security provides an immediate cost and schedule advantage over a new secure policy language or policy distribution protocol.

2.4.3.2.2.2.3 (U) Risks/Threats/Attacks

(U//FOUO) Having a secure policy distribution path will greatly reduce the risk of threats or attacks on the dynamic policy management system.

2.4.3.2.2.3 (U) Maturity

(U//FOUO) Current COTS solutions (e.g., SSL-TLS or IPsec) are very well defined and available. The following products are commercially available today and are candidates for GIG secure policy distribution:

- (U) SSL-TLS Products:
  - (U) F5 Networks Inc., Firepass
  - (U) RSA Security Inc., RSA BSAFE® SSL-J
  - (U) Thawte Consulting (Pty) Ltd , Thawte SSL Web Server Certificate
  - (U) GeoTrust, Inc., QuickSSL® Premium
  - (U) Canfone.com Web Services, eSecure 128-bit SSL Hosting
  - (U) OpenConnect Systems, Incorporated, Secure ClientConnect
  - (U) Citrix Systems, Inc., Citrix MetaFrame Access Suite: Secure Gateway
  - (U) Entrust, Inc., Entrust Authority™ Toolkits
  - (U) Ingrian Networks, Inc., Ingrian i225 - Secure Transaction Platforms
  - (U) VeriSign, Inc., Managed PKI for SSL Certificate
  - (U) Valicert, Inc., Valicert SecureTransport™
- (U) IPsec Products:
  - (U) Check Point Software Technologies Ltd., Checkpoint Secure Platform AI R55
  - (U) DrayTek, Vigor 3300 Version
  - (U) Enterasys Networks, XSR 3000 Series
  - (U) Intoto Inc., iGateway
  - (U) NetScreen Technologies, Inc., NetScreen Security Gateway Product Group
  - (U) Novell, Novell BorderManager
  - (U) Secure Computing Sidewinder G2 Firewall
  - (U) Cisco Systems, Inc., Cisco VPN Client
  - (U) CentricVoice, CentricVoice's IPsec VPN
  - (U) Entrust, Inc., Entrust Authority™ Toolkits.

(U//FOUO) The various sub-technologies of the distribution security technology area can be generally assigned Technology Readiness Level groups of Early, Emerging, and Mature.

- (U//FOUO) COTS SSL-TLS and IPsec products—Mature (TRLs 7 – 9)
- (U//FOUO) Security embedded into policy languages—Early (TRLs 1 –3).

2.4.3.2.2.4 (U) Standards

Table 2.4-5: (U) Distribution Security Standards

This Table is (U)	
Standard	**Description**
SSL	SSL is designed to make use of TCP as a communication layer to provide a reliable end-to-end secure and authenticated connection between two points over a network.
TLS	RFC2246: The primary goal of the Transport Layer Security (TLS) Protocol is to provide privacy and data integrity between two communicating applications. The protocol is composed of two layers: the TLS Record Protocol and the TLS Handshake Protocol. At the lowest level, layered on top of some reliable transport protocol (e.g., TCP), is the TLS Record Protocol. The TLS Record Protocol provides connection security that provides confidentiality and integrity. TLS is designed as a successor to SSL and is sometimes called SSL V3.0.
IPsec	RFC 2401: Internet Protocol Security (generally shortened to IPsec) is a framework of open standards that provides data confidentiality, data integrity, and data authentication between participating peers at the IP layer. IPsec can be used to protect one or more data flows between IPsec peers.
This Table is (U)	

2.4.3.2.2.5 (U) Cost/Limitations

(U//FOUO) A limitation with a COTS solution is how DoD PKI (or other GIG key credentials) would be integrated into COTS products. This assumes that GIG policy distribution would require the use of GIG keys.

2.4.3.2.2.6 (U) Alternatives

(U) The alternative to using COTS security solution for policy distribution would be to develop a secure policy distribution protocol for the GIG system.

2.4.3.2.2.7 (U) References

(U) http://www.faqs.org/rfcs/rfc2246.html

(U) http://www.faqs.org/rfcs/rfc2401.html

(U) http://www.bitpipe.com/plist/SSL.html

(U) http://www.bitpipe.com/plist/IPSec.html

(U)http://www.icsalabs.com/html/communities/ipsec/certification/certified_products/1.0Dindex.s
html

2.4.3.3 (U) Policy Management Architectures

(U//FOUO) One example of a policy management architecture is described in the paper titled "Distributed Multi-National Network Operation Centres" by Scott Shyne (AFRL), David Kidson (CRC), and Peter George (DSTO). This paper describes a coalition network management architecture to use between Australia, Canada, and the U.S. A. This policy-based network management system was developed to manage the coalition domain's network quality of service configuration. The system consists of a domain policy integration manager, policy distribution points, policy enforcement points, and policy delivery protocol. The high level XML policy statements are used to constitute a defined course of action for coalition domains. Each domain must break down the policy into configuration files for use by the network entities for policy enforcement. Local policy is introduced at this level to further define domain operations.

(U//FOUO) Another example is the commercial product SecureSpan, by Layer 7 Technologies. SecureSpan addresses web service security, trust establishment, enterprise policy management, and dynamic policy from the Transport layer through the Application layer. SecureSpan is made up of three major components: SecureSpan Manager, SecureSpan Gateway, and SecureSpan Agent. See http://www.layer7tech.com/products/

- (U) The SecureSpan Manager is a GUI-based application that enables administrators to centrally define, provision, monitor, and audit security and integration policies for Web services.

- (U) The SecureSpan Gateway is a rack-mountable, high-performance network appliance enforces policy on every Web service provisioned through the SecureSpan Manager. The Gateway identifies and processes each message under the policy created for the service. It shields access to internal services, ensuring that only those messages that meet all security and integration policy requirements are forwarded to the destination service.

- (U) The SecureSpan Agent interfaces with client-side applications and automatically negotiates policy-specific security, routing, and transaction preferences with the SecureSpan Gateway.

(U//FOUO) The policy management architecture described in Section 2.4.2 above includes a policy input point, policy repository, policy decision point, and policy enforcement point. A technology that supports the policy repository is a policy directory, as described below.

2.4.3.3.1 (U) Policy Directories

2.4.3.3.1.1 (U) Technical Detail

(U//FOUO) A policy directory can be used as a repository for policies, as well as device information and administrative information needed for policy distribution, deconfliction, synchronization, and promulgation.

(U//FOUO) A directory has several beneficial features that can be used in policy management:

- (U//FOUO) Directories can provide distributed policy management. As the GIG network expands, additional directories can be added to handle new or expanded domains.

- (U//FOUO) Directories also have the ability to shadow or replicate the policy information between policy directories. This capability greatly simplifies the maintenance and management of policy information as policies change or as the network grows.
- (U//FOUO) Directories can also be partitioned to limit access to sensitive data stored in the directory. Partitioning can be configured so that only certain users can have write access to the policy information stored in the directory. Partitioning can also be used to limit read access to only the policies that apply to a specific user or device.

2.4.3.3.1.2 (U) Usage Considerations

2.4.3.3.1.2.1 (U) Implementation Issues

(U) Nortel Networks Optivity Policy Services (OPS) is a software application designed to manage network QoS and network access security. The Nortel OPS product uses a directory as the policy repository. This directory is used to store policies, device information, and related administrative information required by OPS.

(U) Netegrity's SiteMinder product and DMS also use a directory to store critical policy information used in making access control decisions.

2.4.3.3.1.2.2 (U) Advantages

(U//FOUO) The main advantages to using a directory to store GIG policy information are:

- (U//FOUO) Directories have flexible storage schemas to store all types of policy information
- (U//FOUO) Directories have defined interface protocols for access to the data
- (U//FOUO) Directories can limit read and write access to the data
- (U//FOUO) Directories have chaining capabilities that can keep information synchronized between different directories

2.4.3.3.1.2.3 (U) Risks/Threats/Attacks

(U//FOUO) A policy directory would need to be well-protected against improper access to the data stored in the directory. Directories have a binding process where they determine if a person requesting access is who they are and if they should be granted access information stored in the directory.

2.4.3.3.1.3 (U) Maturity

(U//FOUO) Using directories for storing network and system information is very mature. Strong binds and SSL tunnels to directories to make more secure interfaces to the directory data are also in use. There may be additional work needed in the directory access security, depending on the required level of authentication for the GIG program.

(U//FOUO) The various sub-technologies of the policy directories technology area can be generally assigned Technology Readiness Level groups of Early, Emerging, and Mature.

- (U//FOUO) Directory standards—Mature (TRLs 7 – 9)
- (U//FOUO) Directory security—Emerging (TRLs 4 – 6).

2.4.3.3.1.4 (U) Standards

Table 2.4-6: (U) Directory Standards

This Table is (U)

Standard	Description
X.500	X.500 is a CCITT protocol that is designed to build a distributed, global directory. It offers decentralized maintenance, searching capabilities, single global namespace, structured information framework, and a standards-based directory.
Finger, whois, domain name	These are very simple directory formats that are also in use.

This Table is (U)

2.4.3.3.1.5 (U) Alternatives

(U//FOUO) Using a database for the policy repository is an alternative to the directory approach. The database could store all policy information, and a secure interface could be written to control access to the data.

2.4.3.3.1.6 (U) References

(U) http://www.nortelnetworks.com/products/01/optivity/policy/index.html

(U) "The Directory: Overview of Concepts, Models and Service," CCITT Recommendation X.500, 1988.

(U) http://www.netegrity.com/products/products.cfm?page=productsoverview

2.4.4 (U) Dynamic Policy Management: Gap Analysis

(U) Gap analysis for the Dynamic Policy Management Enabler indicates that the main areas of future development are as follows:

- (U//FOUO) Need to further expand the extensible policy languages to cover the complete set of GIG policies. Some existing policy languages such as Ponder, KAoS, Rei, and XACML are flexible in that they allow you to define new policy within the language. GIG should further research these flexible policy languages to see which would be best suited for GIG policies.
- (U//FOUO) Need to develop/refine network modeling and simulation tools used to assess the impact of candidate global and local policy configuration changes on operational risk, network loads and network/application interactions. These policy management testing tools must ensure security requirements for asset usage are not violated. The Ponder toolkit has some capabilities in this gap area.
- (U//FOUO) Need to develop automated policy deconfliction tools. The KAoS policy service and Rei product have some policy confliction resolution capabilities, but these

tools will need to be further developed for the GIG program. Initial versions of this tool may require operator intervention to settle conflicts between policies. As the deconfliction tools mature, this process will become more automated.

- (U//FOUO) Need to develop tools or compilers to translate policy language into a device interpretable language such as a router configuration file. These configuration files are generally vendor specific. Standardizing the end network device configuration formats would greatly simplify this task.

(U//FOUO) Technology adequacy is a means of evaluating the technologies as they currently stand. This data can be used as a gap assessment between a technology's current maturity and the maturity needed for successful inclusion.

(U//FOUO) The Table 2.4-7 lists the adequacy of the dynamic policy management technologies with respect to the enabler attributes discussed in the RCD. Gray entries currently have no technology available, and no research is underway to develop the needed technology. The gray grid entries represent insufficient technology. Solid black entries are adequate today.

Table 2.4-7: (U) Technology Adequacy for Dynamic Policy Management

This Table is (U)

		Technology categories				
		Policy Distribution	**Policy languages**	**Trust Anchor**	**Policy Enforcement Configuration**	**Required Capability (attribute from RCD)**
Enabler Attributes	**Secure solution**					IACNF6, IACNF12, IAINT1, IAPOL6, IAIAC8, IAIAC6, IAIAC9, IACM11, IAAV20
	Standard format					IAPOL8, IAPOL9, IAIAC1, IAAUD7
	Verifiable solution					IACNF15, IAPOL5, IAPOL7, IACM2, IACM4, IACM5
	Policy synchronization and deconfliction					IAAV4, IAPOL1, IAPOL3, IAPOL4, IACM9, IARC08, IARC09

This Table is (U)

2.4.5 (U) Dynamic Policy Management: Recommendations and Timelines

(U//FOUO) The following gaps have been identified in the Dynamic Policy Management Enabler. Without these, this Enabler cannot be fully satisfied. The technology gaps can be of the following types—Standards, Technology, and Infrastructure.

2.4.5.1 (U) Standards

- (U//FOUO) Standards for specifying policy. The policy language needs to cover all GIG policies: access control, quality of protection, quality of service, transport, audit, computer network defense, and policies covering the hardware and software associated with GIG assets. Candidate policy languages include:

    - (U) XACML
    - (U) Ponder
    - (U) KAoS
    - (U) Security Assertion Markup Language (SAML)
    - (U) Rei
- (U//FOUO) Policy deconfliction standard for how to handle policy conflicts
- (U//FOUO) Policy Distribution Standard (push and pull), including protection of policies at rest and in transit, policy validation, distribution error and exception handling
- (U//FOUO) Standard for managing authorities that can promulgate policy and delegate their authority

2.4.5.2 (U) Technology

- (U//FOUO) Mechanisms and performance analysis of policy specification languages and translation to device interpretable language
- (U//FOUO) Performance analysis of various methods of distributing policies (pull and push approaches) to support Policy Distribution Standard
- (U//FOUO) Methods for performing policy synchronization
- (U//FOUO) Tools for analyzing affects of policy and multiple policy objects on overall system
- (U//FOUO) Life cycle model for policy objects
- (U//FOUO) Application of artificial intelligence, heuristics, learning systems, etc., to policy management

2.4.5.3 (U) Infrastructure

(U) Policy management infrastructure that provides:

- (U) Single Graphical User Interface (GUI) for managing multiple classes of assets
- (U//FOUO) Tools for translating automated human language policies into policy base logic
- (U//FOUO) Tools for policy deconfliction
- (U//FOUO) Integrity protection for all policy storage and transfer
- (U//FOUO) Authentication services on all policy exchanges
- (U//FOUO) Logging all policy management transactions

- (U//FOUO) Signed receipts in response to received policy information

(U//FOUO) Figure 2.4-3 contains technology timelines for the Dynamic Policy Management Enabler. These are the results of research completed to date on these technologies. These timelines are expected to evolve as the Reference Capability Document and the research of technologies related to these capabilities continues. The timelines reflect when the technologies could be available given an optimum set of conditions (e.g., commercial community evolution starts immediately, GOTS funding is obtained, staffing is available). Technology topics with missing timelines indicate areas where further work is needed to identify the milestones.

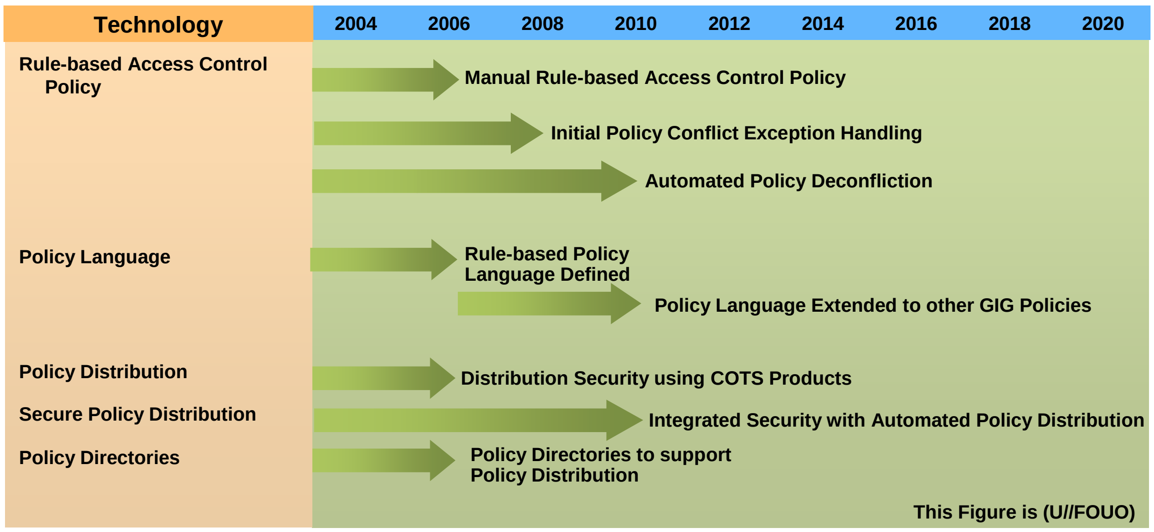


Figure 2.4-3: (U) Technology Timeline for Dynamic Policy Management

2.5 (U) ASSURED RESOURCE ALLOCATION

(U//FOUO) Assured Resource Allocation Enabler maintains the integrity and availability of all enterprise resources (e.g., communication, computing, and core services) and ensures those resources are available to GIG entities—based on operational needs. GIG resources include bandwidth, QoS and priority, processing cycles, access to GIG services, the network management system, routes, and similar assets. Management and allocation of these resources are required for the GIG to meet its operational requirements to provide services to users.

(U//FOUO) This Enabler does not cover the topic of initially designing and implementing the GIG to provide sufficient resources for any end user to accomplish a mission. That is more properly the responsibility of systems engineering and design.

(U//FOUO) This Enabler also does not assume that all GIG users will require resource management services. It assumes the capability needs to exist to deconflict shared resources and to support better-than-best effort service for users that require greater QoS or priority to meet their mission needs.

(U//FOUO) Assured management and allocation includes protecting these management and allocation functions from failures or attacks. It also includes ensuring that no attack or failure can put the GIG into a state where customers cannot get resources to at least the level defined in service level agreements (SLA).

(U//FOUO) Assured Resource Allocation must ensure the availability of computing and communications resources to both GIG infrastructure components and end users. GIG and non-GIG users, processes, and services must not be able to exceed their authorizations and thereby deny or degrade or co-opt services of other GIG users.

(U//FOUO) To meet the GIG 2020 Vision, the GIG architecture must support a number of features. The essential features include:

- (U) Assured Identities
- (U) Digital Access Policy
- (U) IA Policy-based Routing
- (U) Operational-Based Resource Allocation
- (U) RAdAC
- (U) Fault Management, Configuration Management, Accounting Management, Performance Management, and Security Management (FCAPS).

(U//FOUO) These six features are the components of assured management and control of network resources. They combine to provide assurance to the GIG user that requested GIG resources will be available in a securely and equitably managed manner that considers both the nominal/normal privilege status of that user in addition to when the GIG user demand privileges are increased (or decreased) by unique mission or environmental conditions. Their notional interactions may be visualized in Figure 2.5-1.

(U//FOUO) In Figure 2.5-1, the Assured Resource Allocation Enabler acts as a gating function between GIG resources and GIG users. Four of the six components—RAdAC, assured identities, digital access policies, and operational-based resource allocation—act as gate modulators.

(U//FOUO) IA Policy-based routing is a selected or controlled path within the overall path availability to the user. FCAPS has a universal scope of applicability, which means that it impacts all the other five architectural requirements.

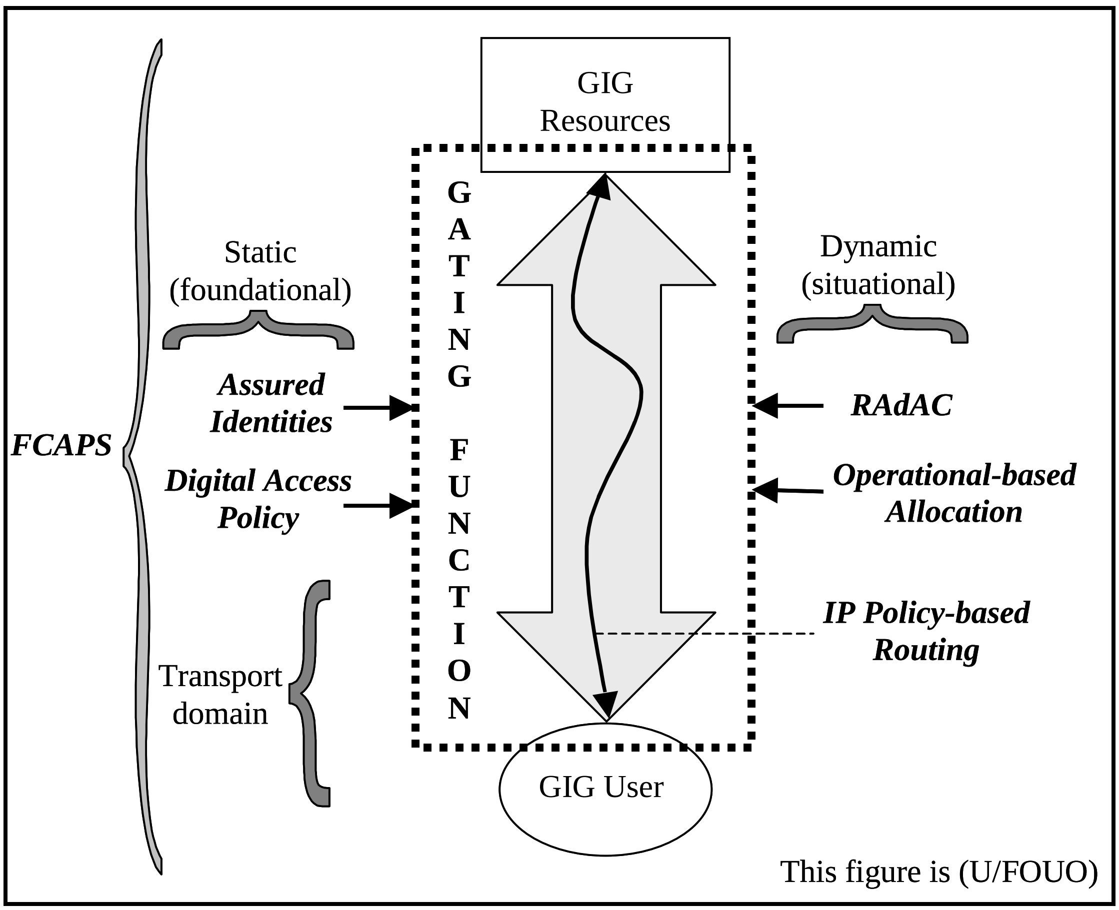


Figure 2.5-1: (U//FOUO) The Role and Components of Assured Resource Allocation

2.5.1 (U) GIG Benefits of Assured Resource Allocation

(U//FOUO) The Assured Resource Allocation Enabler supports continued operation of the system in the face of design failures and hostile attacks. This IA system enabler ensures that there are adequate resources to manage and control the GIG and its attached systems. This enabler applies when:

- (U//FOUO) All data passes through only GIG-controlled systems
- (U//FOUO) Data is transmitted from one portion of the GIG to another through non-GIG controlled systems. GIG management data must also move between portions of the GIG to properly manage resources
- (U//FOUO) User data passes from the GIG to end user systems through non-GIG controlled systems. GIG management data must flow between the GIG and the end system to ensure proper resource management.

(U//FOUO) Assured Resource Allocation provides the following additional benefits to the GIG:

- (U//FOUO) Ensures allocation of GIG resources to meet operational needs (e.g., priority and preemption)
- (U//FOUO) Routes information based upon the specified IA policy, which must account for factors such as Quality of Protection (QoP) for the information, QoS, and priority for the information
- (U//FOUO) Provides enforcement of QoP, QoS, and priority to ensure GIG entities do not exceed their authorizations to deny/degrade service of other GIG users
- (U//FOUO) Provides network control across multiple disparate networks both within the GIG and across both GIG and non-GIG networks
- (U//FOUO) Prevents unauthorized entities from accessing management and control data of the network and network assets

2.5.2 (U) Assured Resource Allocation: Description

(U//FOUO) The GIG core will have a management and allocation system consisting of two major components: the routing and allocation component and the management and control component. Each of the constituent transport programs of the GIG (e.g., Global Information Grid-Bandwidth Expansion [GIG-BE], Transformational Satellite (TSAT), and Joint Tactical Radio System [JTRS]) contains these two components. This fundamental system enabler addresses the IA aspects of these components.

(U//FOUO) The management and control component of the GIG is responsible for monitoring the state of each of the GIG infrastructure components (e.g., communication, computing, and core services) and systems. This component also reacts to changes in the state (e.g., detecting an attack and reacting to it; detecting that a device has failed and taking steps to restart it or route around it).

(U//FOUO) In order to achieve the provisioning of assured management of GIG resources, the following functions must be provided by the GIG:

- (U//FOUO) Transfer of network control (i.e., performance, configuration) across multiple disparate networks (e.g., TSAT, GIG-BE, JTRS) and security domains to support Operational-Based Resource Allocation
- (U//FOUO) QoS/CoS integrity and authorization and priority enforcement mechanisms to ensure that prioritization and precedence requirements are met and to defend against attacks that would allow attackers to hijack or monopolize resources by improperly claiming high priority traffic privileges
- (U//FOUO) Threat-based Traffic Flow Security for network management data to prevent attackers from gaining information about the topology of the network in violation of a system security policy.

(U//FOUO) GIG management and control must function properly for the GIG resource allocation capabilities to be provided. This enabler focuses on assured management that provides protection against attacks on the management and control system.

(U//FOUO) These attacks could take the form of an attacker masquerading as a legitimate management node/user and then modifying a component through the management interface, for example, shutting it down remotely. To prevent this, there must be controlled management and control interfaces. Also, only authenticated components and users should be able to modify a component or the system.

(U//FOUO) In addition, management and control communications should be protected from disclosure to unauthorized individuals. Disclosure of this type of information reveals substantial details about the network topology and capabilities and could provide an attacker a roadmap for a successful attack.

(U//FOUO) The routing and allocation component is responsible for establishing and updating information routing paths as necessary. This includes the initial route establishment, monitoring of the actual flow of data, and the ongoing operation of the routing algorithm to modify paths for changing network conditions (e.g., congestion, failure, attack).

(U//FOUO) IA policy-based routing is essential. The digital policy will stipulate the Quality of Protection required to assure the appropriate security protection is maintained while the data traverses the GIG. This differs from standard commercial networks that use metrics based primarily on cost in their routing algorithms. Routes are chosen to minimize the cost to the service provider and to the end customer of moving bits across the network. Other factors, such as latency or who owns the network or components, are less frequently used.

(U//FOUO) The intent of policy-based routing is to guarantee a minimum level of service to users. This is generally measured in terms of bandwidth (i.e., they will be able to ship X bits per second), latency (i.e., data will take no more than Y seconds to transit from point A to point B), or similar measures. However, GIG routing will also have to factor in the security protection provided by the route and whether this protection is adequate for the QoP required by the data.

(U//FOUO) For security reasons, a low-cost route through a network owned by a coalition partner will often be rejected in favor of a higher cost route through a network owned by the U.S. Government. To meet application requirements, a route with lower latency will sometimes be selected over a lower-cost route with higher latency (e.g., a terrestrial network will be chosen over a satellite connection).

(U//FOUO) Routing decisions of this type constitute IA policy-based routing. The GIG must support this feature. Further, the policy must be changeable for dynamic responses to changing conditions, and the policy must be protected to ensure an adversary cannot substitute or modify a policy to change operation of the GIG.

(U//FOUO) QoS/CoS encompasses designing and implementing a network and its routing infrastructure so that different types (classes) of data are treated differently. Typically, data associated with applications that require real-time delivery with low latency and high likelihood of error-free delivery can be assigned to a class that is forwarded or delivered faster than other traffic, which can be delivered with classic Internet Protocol (IP) best efforts service. Examples of data service applications which require low latency (near real-time), low error rates, and high availability include streaming live video, and real-time collaboration tool services (combining live interactive voice, video, and whiteboarding capabilities), in addition to high quality voice transmissions over IP (VoIP) using high rate voice coders (32 kbps and above). An example of an application that can be delivered with only classic IP best-effort service is e-mail, which can be delivered whenever extra resources are available (typically, any time within the next 5 days). An intermediate data service application that does not require low error rates, but only needs for the low latency and high availability specifications to be met, is secure voice over the FNBDT protocol, whose 2.4 kbps Mixed Excitation Linear Prediction (MELP) vocoder can provide good quality at up to 1% error rates. Thus, depending upon the specific application requirements, a tailored QoS/CoS should be available that meets the desired performance specifications.

(U//FOUO) In order to meet these requirements, the GIG must support certain QoS/CoS mechanisms. However, there is often a clash between QoS/CoS and security requirements. For example, QoS/CoS is often implemented by having the originator indicate to the infrastructure the type of data being sent, so that the core routers can treat it appropriately. However, doing so can result in a leak of potentially sensitive data around an encryption service and provide an excellent covert channel for attackers to use as they wish. Thus, research must be done to develop ways to have the GIG support QoS/CoS and at the same time meet its security requirements.

2.5.3 (U) Technologies

(U//FOUO) The following technology areas support the Assured Resource Allocation Enabler:

- (U//FOUO) IA Policy-Based Routing
- (U//FOUO) Operational-Based Resource Allocation
- (U//FOUO) Integrity of Network Fault Monitoring/Recovery
- (U//FOUO) Integrity of Network Management & Control

(U//FOUO) Since the last two technology areas (Integrity of Network Fault Monitoring/Recovery and Integrity of Network Management & Control) are functionally similar and likely to depend upon the same underlying infrastructures and secure signaling protocols, they will be addressed within the same section.

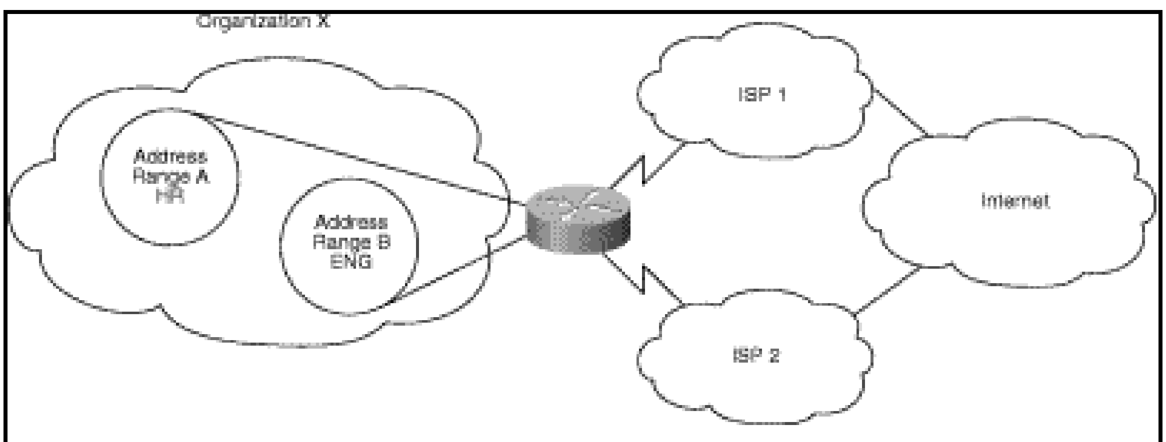
2.5.3.1 (U//FOUO) IA Policy-Based Routing

2.5.3.1.1 (U) Technical Detail

(U//FOUO) Since varying levels of data sensitivity will be traversing the future GIG network routing infrastructure—from unclassified up to and beyond Top Secret—the GIG would benefit from a capability for Information Assurance policy-based routing. To a certain degree, Multi-Protocol Label Switching (MPLS) can provide this attribute. However, MPLS is a static technique that is not amenable to adaptation and dynamic operation in order to react to changing network conditions. Should the network topology change or degrade due to router malfunctions or adversarial denial of service attacks on specific routers, certain predetermined MPLS-Labeled Switch Paths (LSPs) may become similarly broken (if they traverse the affected routers).

(U//FOUO) Any IA policy-based routing scheme should ideally be adaptive and intelligent enough to dynamically react to and compensate for network element outages. In general, IA policy-based routing can be viewed as a subset of QoS–based routing, where the quality being used as a metric happens to be that of information assurance.

(U//FOUO) In very simplistic terms, the Figure 2.5-2 shows an elementary aspect of how IA policy-based routing can be realized:



This figure is (U)

Figure 2.5-2: (U//FOUO) IA Policy-Based Routing

(U//FOUO) As an example, suppose an organization wants to have a subset of its data traffic (traffic of its HR human relations group from address range A) go through Internet Service Provider (ISP) 1 and another subset of traffic (of its Engineering group, from address range B) go through ISP2. It uses different ISPs due to the different sensitivity levels of the two traffic flows and to the commensurate trust put in each of the ISPs. This is an example of Source-Based Transit Provider Selection—Internet service providers and other organizations can use policy-based routing to route traffic originating from different sets of users through different Internet connections across the policy routers.

(U//FOUO) In general terms, Policy-Based Routing (PBR) provides a mechanism for expressing and implementing the forwarding or routing of data packets based on the policies defined by the network policy administrators. It provides a more flexible mechanism for routing packets through routers, complementing the existing mechanism provided by routing protocols. Routers forward packets to the destination addresses based on information from static routes or dynamic routing protocols—such as Routing Information Protocol (RIP), Open Shortest Path First (OSPF), or Enhanced Interior Gateway Routing Protocol (Enhanced IGRP®).

(U//FOUO) Instead of routing by the destination address, policy-based routing allows network administrators to determine and implement routing policies to allow or deny paths based on the following:

- (U) Identity of a particular end system
- (U) Application
- (U) Protocol
- (U) Size of packets
- (U) Security/classification level of traffic data packets
- (U) Security/assurance of links/router nodes

(U//FOUO) Policies can be defined as simply as "My network will not carry traffic from the engineering department." or as complex as "Traffic originating within my network with the following characteristics will take path A, while other traffic will take path B."

(U//FOUO) One of the hallmarks or characteristics of a routing protocol, which enables taking into account the IA aspects of both the routing environment and the data packets that are being routed, is that the protocol must be flexible. This flexibility means different applications can use different paths between the same two points. A mechanism that provides for this capability would include the ability to modify at runtime the routing algorithms and property metrics used to generate forwarding tables. This would essentially result in routers having more than one forwarding table from which to make forwarding decisions, with packets being filtered in order to decide which forwarding table to employ. A routing protocol that utilizes this paradigm is the Flexible Intra-AS Routing Environment protocol (FIRE), developed under the auspices of Defense Advanced Research Projects Agency (DARPA) in 2000. FIRE is an interior gateway-routing protocol that allows traffic to be routed based on a set of routing algorithms rather than one algorithm—such as shortest path first.

(U//FOUO) Today's routing protocols create a single forwarding table for routing decisions. These routing decisions are based on a single configured metric (generally determined by the specifier of the routing protocol, with some modest ability for operators to adjust the metrics). The least cost or shortest path based on that metric is usually what is chosen as the best route.

(U//FOUO) The routing protocols are a closed system—access to routing information is permitted only for participating routers. This is not conducive to modern network architectures where adaptive or active networks provide applications greater freedom to specify the routing services needed. Current routing protocols do not permit applications to actively participate in the routing of their data and make it difficult for researchers and, more importantly, network operators to devise and deploy new metrics such as those they might require for QoS routing.

(U//FOUO) FIRE addresses these problems by substantially enhancing the flexibility of a routing system within an autonomous system. FIRE is a link-state routing protocol, like Open Shortest Path First (OSPF), but rather than advertising a single metric as OSPF does, a FIRE router will advertise a series of property values such as security, cost, and bandwidth. Properties can be configured by an operator, or they can be a value determined at run time. Multiple forwarding tables can then be generated from these properties.

(U//FOUO) In addition, FIRE may use path-generation algorithms other than SPF. For instance, a best path based on highest bandwidth is found by comparing the lowest bandwidth link of all possible paths. Of the lowest bandwidth links, whichever one has the highest bandwidth belongs to the highest bandwidth path. Similar computations would be done if security of specific links were the deciding factor, which would be the case in an IA policy-based routing environment.

(U//FOUO) FIRE separates the routing algorithms from the environment within which these algorithms create forwarding tables. Consequently, the algorithms are treated as applets that are easily installed and replaced. In this respect, FIRE has an Active Networks component for expandability. In general, FIRE would employ a property repository or database for the links/nodes in a subject autonomous system (AS). It would use various routing algorithms, especially tailored to security attributes, to produce forwarding tables. Filters would then be applied to incoming packets to determine which table is appropriate to make a forwarding decision (where various criteria determine the path).

(U//FOUO) A protocol such as FIRE can be implemented because many of the traditional baseline routing protocols have extension capabilities. For example, OSPF and IS-IS allow definition of new state advertisement messages. Thus, FIRE can be viewed as a evolution of the OSPF baseline capabilities.

2.5.3.1.2 (U) Usage Considerations

(U//FOUO) Certain portions of the GIG are likely to require baseline capabilities in support of IA policy-based routing early in the development of the GIG. High Assurance Internet Protocol Encryptor (HAIPE) program products should provide support for routing and QoS by the 2008 timeframe. In addition, the JTRS Wideband Networking Waveform (WNW) program should provide for improved support for route selection, also in the 2008 timeframe.

(U//FOUO) The application of IA policy-based routing techniques may be different depending upon whether the subject portion of the GIG network is wireless (as in JTRS) or wired (as in the GIG-BE core network). Wireless networks naturally are more topologically dynamic than wired networks and, as such, will require more agile IA policy-based routing implementations.

(U) Wireless Applications:

(U//FOUO) There has been some research in the area of IA policy-based routing in tactical wireless communications (as exemplified by mobile ad hoc networks or MANETs). One such study area is Security Aware Ad-hoc Routing (SAR—work done by Yi, Naldurg, and Kravets at the University of Illinois).

(U//FOUO) The SAR protocol operates as follows: When a route of a particular security level is desired, a Route REQuest (RREQ) message is sent out. The RREQ header is encrypted with a group key (known only to those nodes in the network at the same trust level who can handle the desired data security level). The RREQ packet includes a field indicating the overall required route security level. Those intermediate nodes which can decrypt the RREQ then reply with a Route REPly (RREP) message, indicating that they are capable of providing a security guarantee for the path through that node. Thus, eventually, a suitably secure end-to-end path is attained. An advantage of the SAR protocol is that it also provides security to the flow of routing protocol messages themselves.

(U//FOUO) SAR can also be easily incorporated into generic ad hoc routing protocols. In general, SAR enables the automatic discovery of secure routes in a mobile ad hoc environment. Though not optimal, the routes that are discovered by SAR come with quality of protection guarantees. SAR’s integrated security metrics allow applications to explicitly capture and enforce cooperative trust relationships. SAR can be built upon a base routing protocol, such as Ad hoc On demand Distance Vector (AODV), in which case it is known as SAODV.

(U//FOUO) A notional scenario of how this SAR algorithm would operate in tactical applications (using JTRS and/or WIN-T technologies) is depicted in Figure 2.5-3:

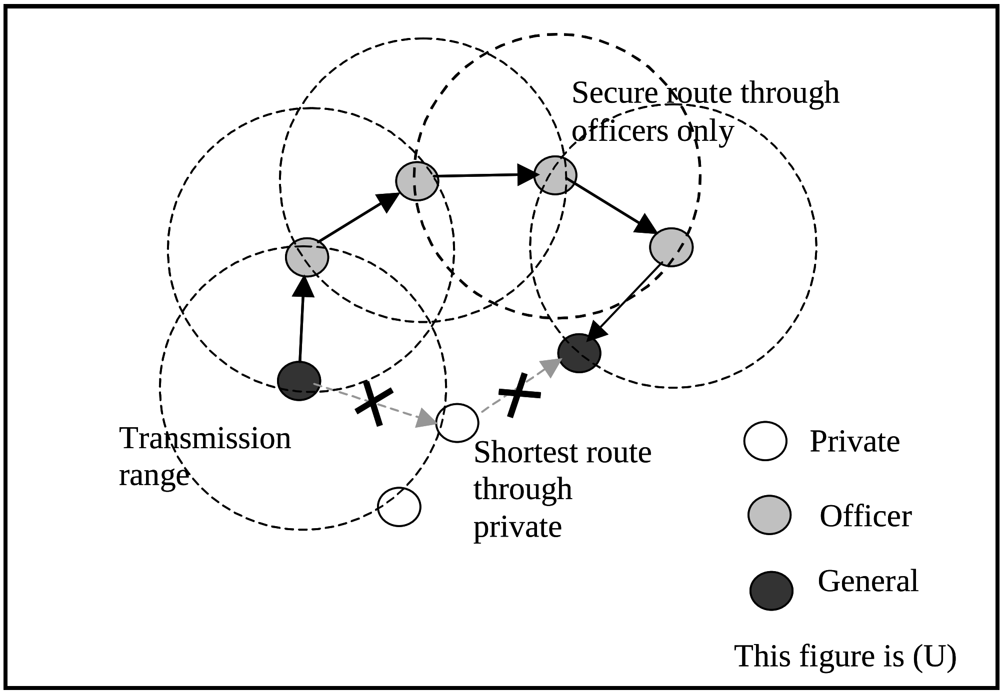


Figure 2.5-3: (U//FOUO) Security-Aware ad-hoc Routing (SAR) in Tactical Wireless Application

(U//FOUO) In the above scenario, even though the second General is reachable most quickly by a path through a Private, the more secure path may be deemed to be only through those with officer rank. The SAR protocol implemented on a tactical Mobile Ad-hoc Networks (MANET) would allow the discovery of the desired path with an appropriate overall integrated end-to-end security metric. Future GIG wireless networks such as JTRS and WIN-T will require similar capabilities so that security attributes can be factored into routing decisions.

2.5.3.1.2.1 (U) Implementation Issues

(U//FOUO) Depending upon the restrictions which are to be imposed upon the core GIG router network, capabilities for full IA policy-based routing may be similarly restricted. For example, in the GIG-BE during its initial implementation phases, there will be no allowance for unprotected information such as QoS levels/specifications to pass from the Red side of the network to the Black side. This potentially limits routing options to static ones, other than routing around any immediately local router node failures that might occur within the Black Core.

(U//FOUO) Fortunately, the HAIPE specification (as written) does make allowance for HAIPE encryption devices to be configured so as to bypass certain information fields (such as QoS bits, IPv6 flow labels, etc) around the encryption process from the Red to the Black domain. However, though many HAIPE encryptors will have this inherent capability, current IA policies tend to prohibit its use due to potential covert channel vulnerabilities. This restriction on an otherwise supported feature is both a GIG-wide implementation issue and a possible limitation to fully dynamic and responsive IA policy-based routing protocols.

2.5.3.1.2.2 (U) Advantages

(U//FOUO) Certainly one of the advantages of a dynamic and flexible IA policy-based routing protocol (as could be implemented within the constructs of the previously described FIRE routing environment) is that it can be automatically adaptive to changing network conditions and topologies. This is compared with static, MPLS path configurations which would not be as survivable or as forgiving to network topology modifications, especially those that would be seen in instances of denial of service attacks. This is due to the fact that MPLS is defined and set up beforehand, by the manual configuration of essentially hard-wired network paths for specific traffic classes. Indeed, the MPLS solution is merely an emulation of a static circuit-switched network solution within the environment of a potentially much more robust and adaptively dynamic packet-switched network fabric.

2.5.3.1.2.3 (U) Risks/Threats/Attacks

(U//FOUO) One of the risks or threats that any network faces is Denial of Service (DoS) or Distributed Denial of Service (DDoS) attacks from adversaries. A good defense of such attacks would include having a routing protocol or mechanism that is dynamic and proactive, in that it would be tied into and integrated with the CND Computer Network Defense/Situational Awareness infrastructure of the subject network. There has been some research into this idea, including some recent work at the University of Arizona ("Impact Analysis of Faults and Attacks in Large-Scale Networks," by Hariri et al, http://dslab.csie.ncu.edu.tw/92html/paper/pdf/Impact%20analysis%20of%20faults%20and%20attacks%20in%20large-scale%20networks.pdf).

(U//FOUO) There is little value in an IA policy-based routing protocol if it only looks at the nominal or normal-condition status of link and nodal security attributes (along with the security characteristics of traffic data packets), without also having means to compensate for either already occurred or impending partial network router fabric failure due to aggressive denial of service attacks. The work at Arizona develops a series of needed metrics, including the Vulnerability Index (VI), Component Impact Factor (CIF), and System Impact Factor (SIF). Using these defined metrics, it then develops a dynamic proactive QoP routing protocol, capable of responding in real time to DDoS router attacks. The primary goal is to maintain availability so that essential network traffic is not denied paths to required end destinations. This is achieved through close observation and analysis of various router operational metrics, such as router buffer utilization, number of flows, and router request-processing rates.

2.5.3.1.3 (U) Maturity

(U//FOUO) Most current routing protocols are based on the policy of finding the shortest path (by application of cheapest cost algorithms) through the given network, for purposes of overall network efficiency and reduction of messaging latency. The extension of routing protocol algorithms to include the aspect or metric of path assurance/security is relatively recent and thus not nearly as mature. Some work in this area has been done for mobile ad hoc networks, due to the obvious potential vulnerabilities of wireless networks as compared with more secure wired network infrastructures. However, some of the ad hoc wireless research results have been extended to the wired domain due to the realization that IA policy-based routing can benefit all networks (wired or wireless).

(U//FOUO) The various sub-technologies of the Integrity of Network Management/Control/Monitoring/Recovery technology area can be generally assigned Technology Readiness Level groups of Early, Emerging, and Mature.

- (U//FOUO) Wireless domain flexible assured routing (SAR, etc.)—Early (TRLs 1 – 3)
- (U//FOUO) Security-driven routing protocols (FIRE, etc.)—Early to low Emerging (TRLs 1 - 4)
- (U//FOUO) Basic MPLS-based (fixed) security routing—Mature (TRLs 7 – 9).

2.5.3.1.4 (U) Standards

(U) Draft U.S. Government Protection Profile on "Switches and Routers" (http://niap.nist.gov/cc-scheme/index.html).

(U) Routing Policy Specification Language (RPSL).

(U//FOUO) There are not many current standards specific to the area of policy-based routing, let alone standards that are devoted to the more specific and delineated area of IA policy-based routing. One standard under development within the IETF is the Routing Policy Specification Language (RPSL). The text of the RPSL specification, as described in IETF RFC 2622, can be found at http://www.ietf.org/rfc/rfc2622.txt (C. Alaettinoglu et. al., 1999).

(U//FOUO) RPSL is merely a language for expressing and conveying routing policies. The language defines a maintainer class (mntner class) object, which is the entity that controls or maintains the objects stored in a database expressed by RPSL. Requests from maintainers can be authenticated with various techniques as defined by the *auth* attribute of the maintainer object. The exact protocols used to communicate RPSL objects is beyond the scope of RPSL as described by RFC 2622, but it is envisioned that several techniques may be used, ranging from interactive query/update protocols to store and forward protocols (similar to email). Regardless of which protocols are used, it is expected that appropriate security techniques, such as IPsec, TLS, or PGP/MIME would be used.

(U) Routing Policy Specification Language next generation (RPSLng):

(U//FOUO) The Internet Engineering Steering Group (IESG) of the IETF has recently initiated work on RPSLng (Routing Policy Specification Language next generation) to add a new set of extensions to RPSL, thus enabling the language to implement routing policies for the IPv6 and multicast address families that are currently used in the Internet. Since the GIG will operate within IPv6 environments (by mandate as of 2008), it is advantageous that RPSL is undergoing this timely updating process. The text of the RPSLng draft can be found at http://www.radb.net/rpslng.txt (L. Blunk et.al., 2004).

(U//FOUO) While the extensions described by RPSLng introduce no additional security threats, it should be noted that the original RFC 2622 describing the RPSL standard included several weak or vulnerable authentication mechanisms. For example, among RPSL-defined mechanisms and constructs, the “MAIL-FROM” scheme can be easily defeated by source email address spoofing. Secondly, the “CRYPT-PW” scheme is subject to dictionary attacks and password sniffing if RPSL objects are submitted by unencrypted channels, such as email. And finally, the “NONE” mechanism option offers no protection for objects.

(U) Related QoS Routing Standards:

(U//FOUO) There are currently several existing IETF RFCs devoted to the description of QoS-based routing mechanisms. IA policy-based routing is merely a specialized subset of QoS-based routing, where the governing QoS is transport security. RFC 2386 “A Framework for QoS-based Routing in the Internet” (Crawley et al, 1998) describes a framework for extending the current Internet routing model of intra and interdomain routing to support QoS.

(U//FOUO) Another relevant IETF standard document is RFC 2676 “QoS Routing Mechanisms and OSPF Extensions” (Apostolopoulos et al, 1999). The GIG is expected to use routing protocols such as OSPF or the related Intermediate System to Intermediate System (IS-IS) protocol.

(U//FOUO) As can be deduced from its name, OSPF normally in its default mode would simply opt to select the shortest path route through a network, without taking into consideration any other metrics such as the security or IA attributes of encountered nodes and links. Fortunately, both OSPF and IS-IS allow modifications of their default operation by the use of extensions, such as the provision to enable definition of new LSA link state advertisement messages (for updating routing tables). As noted in an earlier section, an example of a routing implementation environment that could allow for IA policy-based routing is BBN’s FIRE (Flexible Intra-AS Routing Environment) which takes advantage of the extension provisions within OSPF to enable dynamic and adaptive routing capabilities. Figure 2.5-4 shows how QoS policy-based routing can be implemented within the OSPF core environment:

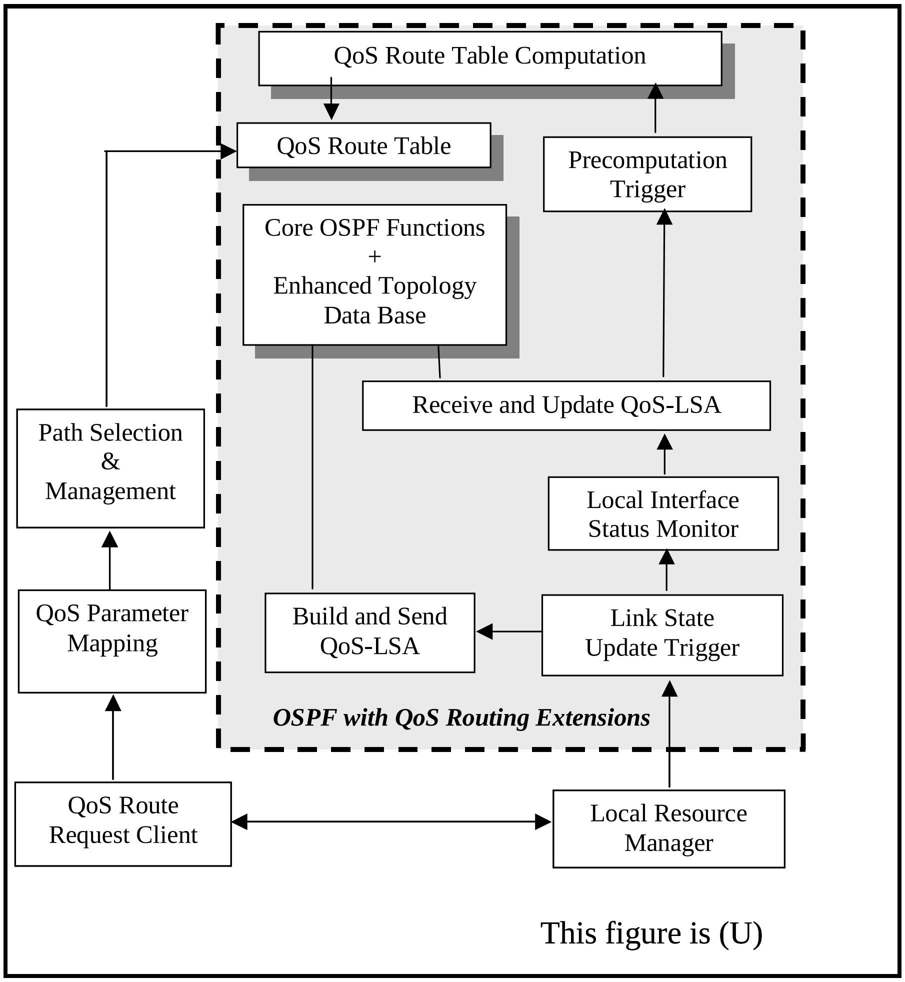


Figure 2.5-4: (U) OSPF Implemented With (QoS) IA Policy-Based Routing Extensions

(U//FOUO) Observation of the above figure shows that such an adaptive and dynamic routing protocol can manage path selection based not only upon metrics such as perceived nominal security of any given links or nodes, but can also factor in such qualities as availability or congestion (based upon the residual bandwidth of network links). Future users of the GIG will not only demand routing based upon assurance but also upon optimized availability.

2.5.3.1.5 (U) Cost/Limitations

(U//FOUO) Any IA policy-based routing methodology will have inherent costs and limitations when implemented in the GIG. Certainly, the installed router software would be more expensive in order to support all of the options presented to a router in so far as assurance-evaluated selectable network paths. Other implied costs would reside in a potential multiplicity of forwarding tables within each router, rather than a single forwarding table per router. Each router would select the relevant forwarding table based upon the IA policy required by the data packet in transit—with more sensitive data choosing the table that yields higher resultant end-to-end assurance levels.

2.5.3.1.6 (U) Dependencies

(U//FOUO) One dependency of the potential evolution and development of a robust, enhanced IA policy-based routing protocol is that it be built upon the foundation of an extensible baseline protocol. One such protocol which allows for extensibility is the OSPF protocol, which is related to the IS-IS protocol, both of which the GIG is likely to use.

(U//FOUO) Another dependency of the development of a robust IA policy-based routing protocol for the future GIG network is that of the required foundation of a GIG standard for Quality of Protection (QoP). Given a QoP definition, whereby specific data entities or packets are to be tagged with information (metadata) that marks the packets for handling and routing tailored to the sensitivity of the data contents, an IA policy-based routing protocol can then use the QoP metadata to optimize the overall network security of the various traffic flow elements.

2.5.3.1.7 (U) Alternatives

(U//FOUO) As has been already noted, an alternative to a fully implemented IA policy-based routing protocol is the use of static MPLS routing. Although this is not as effective and flexible (or fine-grained) a solution as a dynamic policy-based one, it is better than having no provision at all for the protection of sensitive data classes.

2.5.3.1.8 (U) Complementary Techniques

(U//FOUO) In addition to being seen as an alternative solution, there is no reason why MPLS cannot be used in conjunction with (or within the context of) a larger framework of an IA policy-based routing methodology.

2.5.3.1.9 (U) References

(U) "Routing Policy Specification Language (RPSL)," by Alaettinoglu et al, http://www.ietf.org/rfc/rfc2622.txt , or http://www.faqs.org/rfcs/rfc2622.html , 1999.

(U) "QoS Routing Mechanisms and OSPF Extensions," by Apostolopoulos et al, http://www.ietf.org/rfc/rfc2676.txt , or http://www.faqs.org/rfcs/rfc2676.html, 1999.

(U) "A Framework for QoS-based Routing in the Internet," by Crawley et al, http://www.ietf.org/rfc/rfc2386.txt , or http://www.faqs.org/rfcs/rfc2386.html , 1998.

(U) "Integrating Quality of Protection into Ad Hoc Routing Protocols," by Yi, Naldurg, & Kravets, http://mobius.cs.uiuc.edu/publications/sci02.pdf , 2002.

(U) "Security-Aware Ad-Hoc Routing For Wireless Networks," by Yi, Naldurg, & Kravets, http://www.csee.umbc.edu/courses/graduate/CMSC628/spring2002/ppt/poonam.ppt, 2002 (talk at UMBC).

(U) "Security Aware Ad-hoc Routing (SAR)," by Yi, Naldurg, & Kravets, http://www.cs.fsu.edu/~yasinsac/group/slides/carter5.pdf, 2002.

(U) "Routing with Confidence: Supporting Discretionary Routing Requirements in Policy Based Networks," by Kapadia, Naldurg, & Campbell, http://choices.cs.uiuc.edu/~akapadia/papers/sec_routing.pdf.

(U) “FIRE: Flexible Intra-AS Routing Environment,” by Partridge et al, http://www.cs.ndsu.nodak.edu/~yizhang/Presentations/FIRE.ppt, 2000.

(U) http://www.ir.bbn.com/projects/fire/architecture/architecture.html

(U) http://www.ir.bbn.com/documents/techmemos/

(U) http://www.ir.bbn.com/documents/techmemos/TM1245.pdf

(U) http://www.ir.bbn.com/documents/techmemos/TM1244.pdf

(U) http://www.ir.bbn.com/documents/techmemos/TM1265.pdf

(U) http://www.ir.bbn.com/documents/articles/FIREjsac-3-01.pdf

(U) “Impact Analysis of Faults and Attacks in Large-Scale Networks,” by Hariri et al, http://dslab.csie.ncu.edu.tw/92html/paper/pdf/Impact%20analysis%20of%20faults%20and%20attacks%20in%20large-scale%20networks.pdf (also see http://dslab.csie.ncu.edu.tw/92html/paper/ppt/Impact%20analysis%20of%20faults%20and%20attacks%20in%20large-scale%20networks.ppt), 2003.

(U) “Intelligent Active Routing For Supporting QoS Demands,” by Tasir et al, http://www.cntr.salford.ac.uk/telecoms_research/research_activity/abdul_rahman_mohamed_tasir/pgnet2002.htm , 2002.

(U) “Security Cost Routing,” by Eli Winjum, http://web.unik.no/users/mobkom/presentasjoner/Eli_SecurityCost_250803.ppt, 2003.

(U) “Policy-Based Routing,” http://www.cisco.com/warp/public/cc/pd/iosw/tech/plicy_wp.htm, 2002.

(U) “Configuring Policy-Based Routing,” http://www.cisco.com/univercd/cc/td/doc/product/lan/cat4000/12_1_19/config /pbroute.pdf

(U) “QoS/Policy/Constraint-Based Routing,” by Wei Sun, http://www.cse.ohio-state.edu/~jain/cis788-99/ftp/qos_routing/, 2000.

2.5.3.2 (U//FOUO) Operational-Based Resource Allocation

2.5.3.2.1 (U) Technical Detail

(U//FOUO) The technical area of operational-based resource allocation is predominantly one of the pure research realm, with fairly few examples of fielded systems that employ this capability (in an automated sense). There are very few commercial efforts in this area—with the common response to the assurance of adequate resources being that of initial over-provisioning of computation and/or transport assets, so that all potential customers will be adequately served. However, in the defense/military field, there has been some research efforts dedicated, most recently sponsored by a variety of DARPA programs. Future customers of the GIG will expect and demand certain levels of network transport, database access, and computational services. Each customer will have a dynamic/changeable user profile that will describe the privileges that are given to that customer. The future GIG Privilege Management Infrastructure (PMI) will necessarily work very closely with a resource allocation system tailored to customer-centric operational demands.

(U//FOUO) A traditional example of operational-based resource allocation is the Multi Level Precedence and Preemption (MLPP) mechanism that has been used for years in the context of the DoD voice telecommunications system. It is desirable to have the MLPP paradigm, which is nominally only for voice communications control/allocation purposes, extended to the packet-switching and enterprise services-based GIG environment. This extends the MLPP paradigm to coverage of far more system functionality.

(U//FOUO) As is implied in the MLPP acronym, this paradigm allows for an a priori allocation through the precedence route of the (limited) resource of a telecommunications link to a customer whose rank or privileges exceed those of other potential service customers. Precedence decisions are made before the link is fully established. Thus this is a somewhat static and non-adaptive process. In addition, however, the preemption process of MLPP enables an already allocated resource of a telecommunications link to be taken away from the initial customer, or preempted, and to be given to a customer with higher privileges and immediate requirements. Hence, the preemption process is more dynamic and agile than precedence. Both of these capabilities—precedence and preemption—would be useful within the context of the GIG in terms of allocating data transport, data storage, computation, and enterprise service capabilities.

(U//FOUO) Current DoD Information Resources Management (IRM) is fairly inconsistent in its mechanisms for the allocation or re-allocation of communications and other services. Each separate network service or layer has its own mechanism: circuit switched (voice) uses the MLPP protocol, satellite circuits are allocated according to the priorities defined in Chairman of the Joint Chiefs of Staff Instruction (CJCSI) 6250.01, and the current common user data networks have little or no priority-based assignment capabilities.

(U//FOUO) Rather than have a similarly disjoint solution in the future GIG environment—where the resources of Transformational Satellite (TSAT), GIG-BE routers, JTRS nodes/links, and NCES services will all be interacting with each other—a common and integrated resource allocation solution is required. This solution will be required to span across the boundaries of the various GIG systems. Until now, however, many DoD and Intelligence Community (IC) networks have avoided the implementation of automatic allocation and re-allocation mechanisms by implementing community of interest (COI) networks that are small enough to allow for effective manual arbitration. The efficiency-driven use of a common GIG infrastructure will force the DoD and IC to address this issue of enterprise-wide automatic resource allocation.

(U//FOUO) Several programs under the auspices of DARPA have studied the area of dynamic and operational-based resource allocation over five years. These include the following:

- (U) QUORUM Project
- (U) Agile Information Control Environment (AICE) Program
- (U) Battlefield Awareness and Data Dissemination (BADD) Program.

(U//FOUO) One methodology for the automation of dynamic operational-based resource allocation, developed under DARPA auspices, is that of Dynamic Scalable Dependable Real-Time Systems (DeSiDeRaTa). Figure 2.5-5 shows the basic ideas behind DeSiDeRaTa:

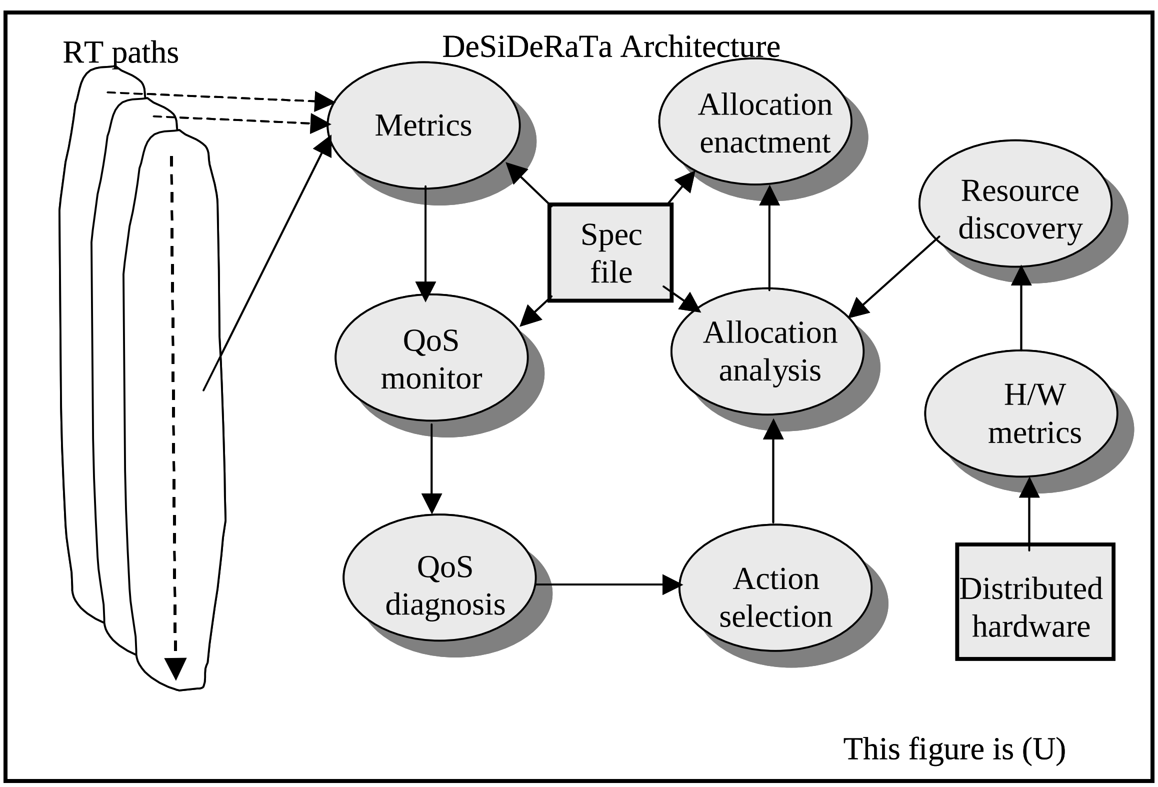


Figure 2.5-5: (U) DeSiDeRaTa Architecture for Operational-Based Resource Allocation

(U//FOUO) From the above figure, it can be seen that DeSiDeRaTa is divided into three vertical groups of functions. The left group deals with QoS measurement and analysis, the central deals with allocation analysis and actions, and the right deals with resource analysis (or resource situational awareness). This model could be applicable to the GIG where resource allocation and re-allocation decisions would be made by adjudication of resource requests against the applicable customer privilege profiles (managed within the GIG's PMI. Overall control of the DeSiDeRaTa mechanisms would be managed by using a nominal specification file, which would consist of the desired (and allowed) customer QoS and the translatable and relevant, required resources. These resources would consist of GIG transport, computation, data storage, and enterprise services access.

(U//FOUO) The next generation of computing and networking is leaning heavily towards the paradigm of distributed computing and networking. As distributed real-time systems—such as those that will be found within the GIG—become increasingly popular, there is an increasing need of technology that can handle the resource allocation problems presented by distributed computing and networking. It is from this basis that DeSiDeRaTa Resource Management has found a grasp in the research community. The DeSiDeRaTa project has the goal of producing a Resource Manager that provides the following features:

- (U) Specification Language for Hardware Systems, including computing resources and networks
- (U) Specification Language for Software Systems, including methods of specifying QoS requirements such as real-time, scalability, and dependability QoS constraints
- (U) QoS Management for instrumentation, assessment, prediction, negotiation, and allocation of resources for real-time systems.

(U//FOUO) DeSiDeRaTa technology will employ the dynamic path paradigm, which is a convenient abstraction for expressing end-to-end QoS objectives of systems and for performing QoS management. The DeSiDeRaTa project provides an adaptive resource management approach that is appropriate for systems (such as the GIG) that expect to experience large variations in workload. A distributed collection of computing resources is managed by continuously computing and assessing QoS metrics and resource utilization metrics that are determined a posteriori.

(U//FOUO) The DeSiDeRaTa project's specification language describes the environment-dependent (and operationally-driven) features of dynamic real-time systems. Also provided is an abstract model that is constructed (statically) from the specifications, and is augmented dynamically with the state of operational environment-dependent features. The model is being used to develop algorithms for QoS monitoring, QoS diagnosis, and resource allocation analysis. Experimental results show the effectiveness of the approach for specification of real-time QoS, detection and diagnosis of QoS failures, and restoration of acceptable QoS by re-allocation of distributed computer and network resources.

(U//FOUO) Future GIG customers who are given temporary privileges for access to certain GIG resources due to operational exigencies would benefit from the dynamic real-time checking that this protocol potentially affords, so that quality of service levels would be maintained and adjusted to satisfy operational requirements. In this sense, DeSiDeRaTa can be viewed as being simultaneously Proactive and Reactive in its methodology for the allocation and re-allocation of resources (see http://www.atl.external.lmco.com/overview/papers/1117.pdf). The DARPA Quorum project analyzed the applicability of DeSiDeRaTa for proactive and reactive resource allocation.

(U//FOUO) Besides the potentially relevant DeSiDeRaTa protocol, there have been other projects done under DARPA auspices in the area of dynamic requirements-driven resource allocation. However, as already noted, this is a relatively new field with few fully mature implementations. Most instantiations of resource allocation to date are manually configured, as opposed to policy-driven automatic implementations, which is the desired end-state of the GIG.

(U//FOUO) Research done during 2001 by a team at Colorado State University (CSU) (under the auspices and sponsorship of the DARPA AICE and BADD programs) concentrated on operational-based dynamic resource allocation for classes of prioritized session and data requests in preemptive heterogeneous networks (http://www.engr.colostate.edu/~echong/pubs/conf/pdpta01.pdf). The GIG can be viewed as such a large heterogeneous network, and certain classes of data within it will be prioritized—based upon the operation of the GIG standard for precedence and preemption.

(U//FOUO) The work done at CSU could potentially be relevant to the internal specifics of this GIG foundational standard. CSU defined network transactions (or communication requests) as one of either two types: Data or Session (session being defined as bandwidth access over a certain timespan). Furthermore, network requests are assigned to a Class and a Priority level (within the class). For purposes of precedence analysis, the request ‘worth’ is computed as a weighted priority that is a function of the situation (war time, peace time, etc.). The CSU methodology then devises a scheduling heuristic that reorders customer service requests by maximizing the sum of weighted priorities of the highest class and then works down the class hierarchy.

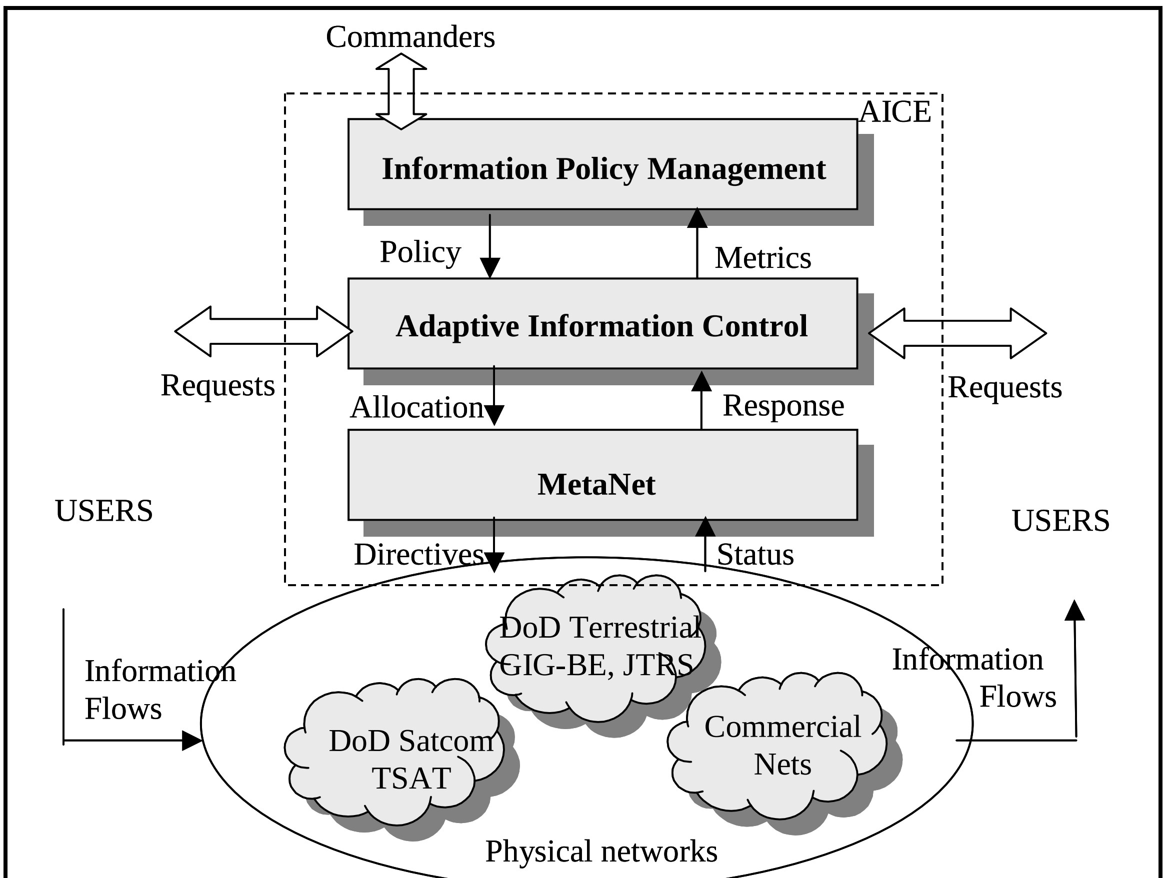
(U//FOUO) An important issue raised by the CSU researchers is the need for a post-preemption scheduler so that any transaction request which is preempted is not lost but is rationally rescheduled in a logically prioritized sense. This rescheduling mechanism can be relevant to the development of a GIG Precedence and Preemption standard.

2.5.3.2.2 (U) Usage Considerations

2.5.3.2.2.1 (U) Implementation Issues

(U//FOUO) Any operational-based resource allocation system in the future GIG infrastructure must have the capability for dynamic modification of customer privilege profiles within the PMI. Future military commanders will not always require privileges that consistently and persistently put them at the head of the line when it comes to getting requested resources before others. Only at times when unique and specific operations are underway will it be necessary for participating individuals to have their privilege status elevated. When the subject operation is completed, participating GIG customers shall in all likelihood have their privilege status relegated and re-baselined back to their normal levels. Since this implies a dynamic privilege management infrastructure, it is important that the PMI be robust and secure, and that the necessary policy adjudication entities be present to authorize any temporary modifications or elevations of privileges.

(U//FOUO) Operational-based resource allocation can be viewed as an exercise in adaptive information control across a distributed landscape. As such, the DARPA AICE Program has conducted a number of relevant studies. The GIG landscape consists of a number of interconnected disparate networks (TSAT, terrestrial wired GIG-BE, wireless JTRS, and WIN-T, etc.) over which resources will be allocated. The transport networks themselves are also allocated resources (for the transport of user communications, sensor data, database query results, and enterprise services, etc). There is a need for the study of the global/overall control and allocation of these disparate network resources so that the integrated services provided to the subject customer base are maximized and optimized. The following figure (based upon work for DARPA by S. Jones and I. Wang of Johns Hopkins Applied Physics Lab) (http://www.engr.colostate.edu/~echong/pubs/conf/00985799.pdf) illustrates a partitioning of the required signaling to achieve joint resource allocation across disparate networks.



This figure is (U)

Figure 2.5-6: (U) Joint Resource Allocation Across GIG Networks(U//FOUO) The above illustration separates the operational-based resource allocation functions into four different but interconnected layers:

- (U//FOUO) Physical Network Layer
- (U//FOUO) This layer consists of the independent tactical (JTRS), terrestrial (GIG-BE),

satellite (TSAT), and wireless, and commercial (Internet) networks that will together comprise the end-to-end user GIG fabric. They will provide packet routing services and unique QoS capabilities.

- (U//FOUO) MetaNet Layer

- (U//FOUO) This layer is the system that facilitates the QoS-based routing through the integrated collection of networks. Four aspects of the MetaNet layer include: inserting QoS-like capabilities into existing tactical networks to enable dynamic (and operational-based) re-allocation of network resources, negotiating service requests as an intermediary between the user and individual networks, providing end-to-end QoS solutions within a time-constraint, and maintaining negotiated end-to-end QoS by dynamically re-routing or renegotiating service.

- (U//FOUO) Adaptive Information Control (AIC) Layer

  (U//FOUO) This layer provides global content-aware dynamic information flow control, employing the services of the MetaNet layer to do so. AIC layer features include: partitioning of information flows among available logical channels, globally optimizing allocation to achieve military users' information flow priorities (precedence and preemption), and re-allocating resources when necessary due to network QoS degradation.

- (U//FOUO) Information Policy Management (IPM) Layer

  (U//FOUO) This layer has three primary functions: providing users the capability to visualize the impacts of their information control policies, relating information policy management to military operations, and aiding in the synthesis of effective information control policies. It is from this layer that relevant and temporary modifications to GIG customer privilege profiles will be made (within the GIG privilege management infrastructure), whereby users are allocated the resources sufficient to successfully conduct military operations.

(U//FOUO) The ultimate objective of the DARPA AICE program is to realize information control and resource allocation in a way that is faster, more efficient, and more precise than is currently realized—(and in an automatic fashion as opposed to manually.

2.5.3.2.2.2 (U) Advantages

(U//FOUO) Certainly one of the advantages of a well constructed operational-based resource allocation system within the future GIG environment is that the overall operation and congestion of the GIG can be optimized to service the most important needs at any given time and in any given theatre of operations. This implies that an alternative over-provisioning solution need not be required. This thus yields savings in the fielded network infrastructure (transport, storage, and computational) equipment. This can especially be true in the case of wireless segments of the GIG (such as mobile ad hoc networks within the JTRS and WIN-T networks), where the network ‘mesh’ is topologically dynamic and potentially sparse.

2.5.3.2.2.3 (U) Risks/Threats/Attacks

(U//FOUO) Since resource allocation will be based upon the privileges of requesting GIG customers, it is essential that both the specific resource requests and the customer privileges be secure, trusted, and not subject to tampering or modification by adversaries.

2.5.3.2.3 (U) Maturity

(U//FOUO) Maturity of operational-based resource allocation technology is fairly low level, especially resource allocation that is automatic as opposed to manual (and human operator intensive). Resource allocation traditionally has been limited to the scope of small geographic areas, as opposed to the world-wide reach of the GIG network.

(U//FOUO) Future warfighters in 'hot-spots' who require and deserve unique privileges to resource access will need to have special consideration in the allocation of GIG transport, computation, storage, and database access capabilities. All of these GIG resources will be distributed. It is the coordination and timely delivery of these resource capabilities that will need research and study before this technology area can be said to be in any stage of maturity.

(U//FOUO) The various sub-technologies of the Integrity of Network Management/Control/Monitoring/Recovery technology area can be generally assigned Technology Readiness Level groups of Early, Emerging, and Mature.

- (U//FOUO) MLPP in Defense Information System Network (DISN) voice telecommunications—Mature (TRLs 7 – 9)
- (U//FOUO) Adaptive/Dynamic distributed resource allocation (like DeSiDeRaTa)—Emerging (TRLS 4- 6)
- (U//FOUO) Operational resource allocation tied to secured/adaptive PMI—Early (TRLs 1 – 3).

2.5.3.2.4 (U) Standards

(U//FOUO) Since there are few commercial or industrial efforts in this technology area (such as by the IETF), there are not any real standards relevant to operational-based resource allocation. As the technology is developed, standards (within the GIG community) should be commensurately developed, so as to assure that all participants within the GIG would be using the same protocols. As a corollary to the implementation of standards for the actual mechanics of resource allocation or re-allocation, a parallel, supporting GIG standard will be needed for Precedence and Preemption (as a subset of the overall GIG privilege management infrastructure).

2.5.3.2.5 (U) Cost/Limitations

(U//FOUO) Any operational-based resource allocation system for the future GIG will have to be cognizant of the possibility that instantaneous local demands in any potential future theatre of operations may exceed the possible delivery capacity (in terms of transport throughput, etc.). As such, methodologies and technologies that are developed must have built-in mechanisms for intelligent resource trimming and notification and also for intelligent policy-driven arbitration in cases of simultaneous demands by disparate customers for the access to the same common resources.

2.5.3.2.6 (U) Dependencies

(U//FOUO) Successful implementation of operation-based resource allocation within the GIG will be dependent upon a number of other developments, especially that of the development of a GIG-wide standard for priority and preemption capability. This standard would be required to clearly define the priority status levels and classes in which all GIG customers will be assignable, in addition to the mechanisms for modifications (and reversions to nominal levels) of user privileges.

2.5.3.2.7 (U) Alternatives

(U//FOUO) An alternative to the necessity of developing an operational-based resource allocation capability within the future GIG is merely to have over-provisioning of required assets (computational, storage, and transport) across the future GIG. While this may be a potential solution when viewing across the GIG as a whole, it will probably not succeed when specific local assets are exceeded by temporarily excessive local demands (as could be the case in a theatre of war). As such, the GIG will require an allocation system that will provide priority claims on assets for those with the highest adjudicated, locally-valid privileges.

2.5.3.2.8 (U) Complementary Techniques

(U//FOUO) Complementary, or subsidiary, techniques for the operational-based allocation of resources include those of the traditional MLPP techniques currently used in the circuit-switched DoD DISN voice network. There are current efforts under pursuit by DISA to fully implement MLPP capabilities within the future GIG-BE router mesh fabric, where voice will no longer be circuit-switched but will instead be VoIP. This IP version of MLPP capabilities should be viewed as part of the future integrated overall resource allocation/re-allocation infrastructure of the GIG—all driven by an underlying dynamic and secure privilege management infrastructure.

2.5.3.2.9 (U) References

(U) "Proactive and Reactive Resource Allocation," by Cross & Lardieri, http://www.atl.external.lmco.com/overview/papers/1117.pdf, 2000.

(U) "A MetaNet Architecture For End-To-End Quality Of Service (QoS) Over Disparate Networks," by Jones & Wang, http://www.engr.colostate.edu/~echong/pubs/conf/00985799.pdf, 2001.

(U) "A QoS Performance Measure Framework for Distributed Heterogeneous Networks," by Kim et al, http://www.cs.nps.navy.mil/people/faculty/irvine/publications/2000/empdp00_QoSMSHN.pdf, 2000.

(U) "Applying Intent-Sensitive Policy To Automated Resource Allocation: Command, Communication and Most Importantly, Control," by Funk et al, http://www.sift.info/English/publications/HICS_AICE-00.pdf, 2000.

(U) "Dynamic Resource Allocation for Classes of Prioritized Session and Data Requests in Preemptive Heterogeneous Networks," by Naik et al, http://www.engr.colostate.edu/~echong/pubs/conf/pdpta01.pdf, 2001.

(U) "Information Value based Information Resource Management of the Defense Information Systems Network," by Devens & Pitt, http://www.argreenhouse.com/society/TacCom/papers99/08_5.pdf , 1999.

(U) "Secure-RM: Security and Resource Management for Dynamic Real-Time Systems," bu Tjaden et al (Ohio University), http://jarok.cs.ohiou.edu/papers/scc2000.pdf, 2000.

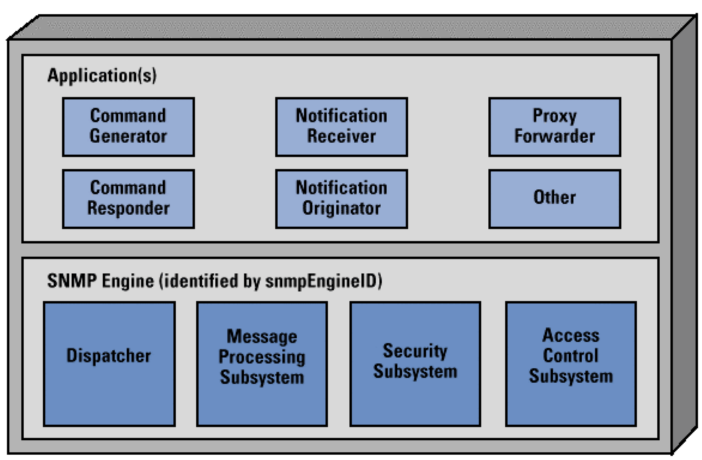
2.5.3.3 (U//FOUO) Integrity of Network Fault Monitoring/Recovery and Integrity of Network Management & Control

2.5.3.3.1 (U) Technical Detail

(U//FOUO) One of the most important IA aspects of the future GIG will be that of securely managing and controlling—both locally and remotely—the various and many network elements. On top of this, should portions of the GIG infrastructure become impaired due to an external attack, component failure, or malfunction, there would be a need for robust and distributed, network fault monitoring and recovery. Since all of these functions rely upon a well-defined set of common sensing (incoming) and command (outgoing) message constructs, a standardized protocol such as the IETF Simple Network Management Protocol (SNMP) would provide the required capabilities. SNMP is a default standard methodology for network management and has survived numerous competing standard entrants.

(U//FOUO) What is really required for successful network management, control, and monitoring, is an entire framework built around three foundation components: a data definition language as defined by an Internet-standard Structure of Management Information (SMI), a set of definitions of management information as delineated by an Internet-standard Management Information Base (MIB), and a common protocol definition (SNMP). The MIB database resides generally at the managed client/agent, and its variables define the scope, range and limitations of control features which may be executed. The SNMP protocol is used to convey information and commands between network managers and managed objects (or agents).

(U//FOUO) There are four basic operations or commands that may be executed within the SNMP protocol. These are Get, GetNext, Set, and Trap. The first three commands are initiated by the manager, and they act upon MIB variables at the client agent of interest. The Trap message is initiated by a client agent when an error or fault occurs, and it is used in order to notify the central manager that something unexpected has gone wrong. The basic elements of SNMP operation are shown in Figure 2.5-7.



This figure is (U)

Figure 2.5-7: (U) Basic Elements of SNMP Operation

(U//FOUO) Note the security-relevant components of the Security Subsystem and Access Control Subsystem in the above figure. It is these component elements that have evolved considerably during the evolution of SNMP through its SNMPv1, SNMPv2, and SNMPv3 versions.

(U//FOUO) The first two versions of SNMP had no real security functionality. Security was primarily introduced in the SNMPv3 implementation. Both authentication and privacy capabilities were introduced by SNMPv3, as shown in Figure 2.5-8.

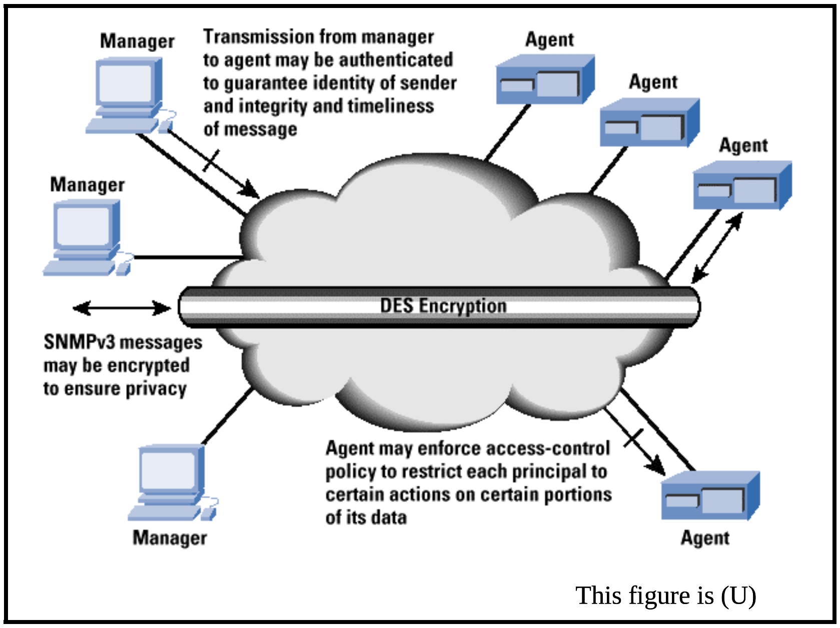


Figure 2.5-8: (U) SNMPv3 Security Capabilities

(U//FOUO) The User Security Model (USM) describes operations of the security functions within SNMPv3. In the basic model, cryptographic keys are assumed to be symmetric or private keys. Authentication is accomplished by using Hashed Message Authentication Code-Message Digest Algorithm 5 (HMAC-MD5) or alternatively HMAC- Secure Hash Algorithm 1 (SHA-1). Encryption or message privacy is accomplished using the Digital Encryption Standard (DES) in the Cipher Block Chaining (CBC) mode. The SNMPv3 message format, as implemented with USM, along with the application scopes of authentication and encryption, is shown in Figure 2.5-9.

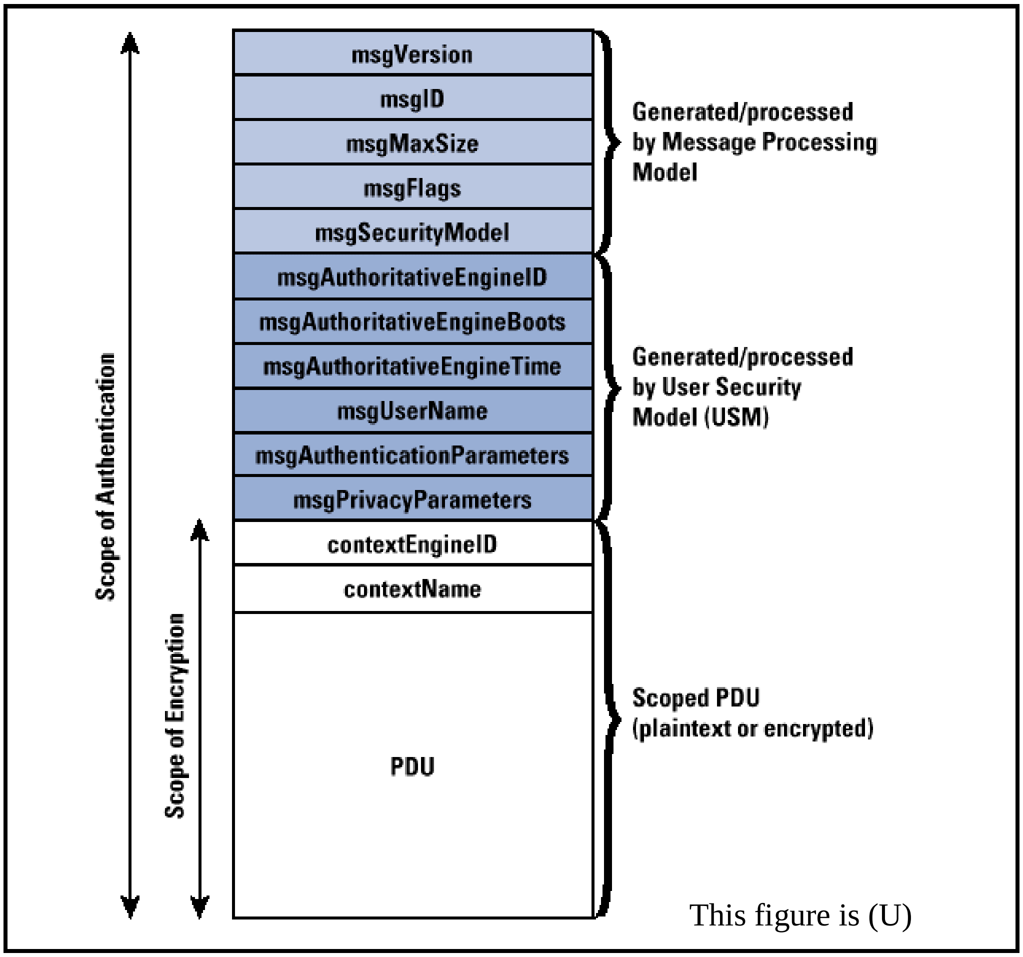
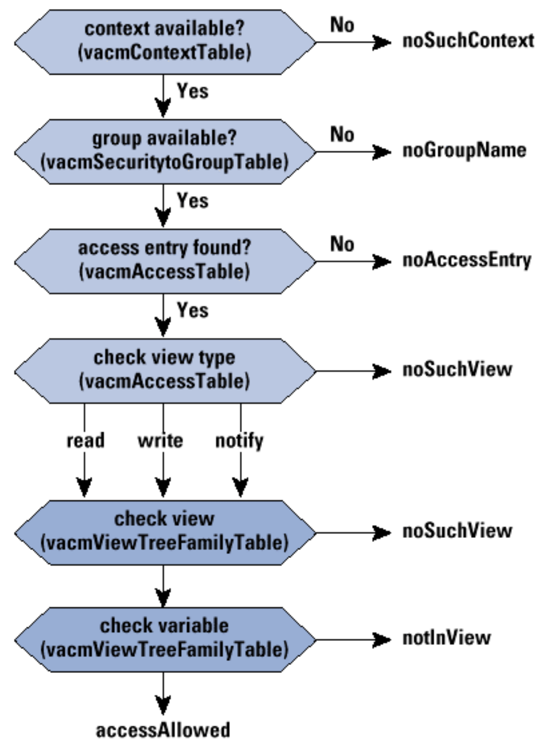


Figure 2.5-9: (U) SNMPv3 Message Format & Security Components

(U//FOUO) The MD5 message digest algorithm (or optional SHA1) indirectly provides for data origin authentication, and it directly defends against data modification attacks.

(U//FOUO) One of the important security features of SNMPv3 is the View-based Access Control Model (VACM) that it employs. VACM determines whether access to a managed object or agent should be allowed. To do this, VACM makes use of an MIB that defines the access control policy for the subject agent—thus enabling remote configuration capabilities. VACM is flexible in that its logic provides for access to be decided by a series of relevant questions concerning the access request: "Who ? + Where ? + How ? + Why ? + What ? + Which ?". Based on the answers to these questions, in conjunction with the contents of policy-based access tables, access is either allowed or disallowed. Figure 2.5-10 shows the access control logic employed by VACM:



This figure is (U)

Figure 2.5-10: (U) SNMPv3 View-based Access Control Model (VACM) Logic

(U//FOUO) The addition of the VACM capability within SNMPv3 should enable future GIG applications to conduct policy-based and fully access-controlled remote and distributed network management and monitoring functions. As such, it is a powerful construct.

2.5.3.3.2 (U) Usage Considerations

(U//FOUO) Some components of the future GIG have already proposed using SNMPv3 in order to enable the IA of management control and monitoring functions. For example, the TSAT program proposes the use of SNMPv3 for network monitoring (as mentioned on slide 23 of the briefing “TCM IA Architecture Overview”, 30 June2004, by NSA’s IAD TSAT IA Integrated Program Team [IPT]). Network management and control of the TSAT network will be required to be at the MAC I level (Mission Assurance Category), the highest of the three defined MAC levels (for a system requiring high integrity and high availability). Similarly, TSAT network management and control will require a confidentiality level of “Sensitive” (or the medium of 3 possible confidentiality levels). The SNMPv3 protocol is deemed adequate in satisfying network monitoring requirements.

(U//FOUO) Many experts (for example, computer science professor Dr. Richard Stanley of Worcester Polytechnic University) have said that SNMPv3 is the “clear long-term choice” for secure network management. Unfortunately, SNMPv3 is still a work-in-progress even within the IETF standardization process. SNMPv1 still holds 95% of the commercial market, with even the intermediate SNMPv2 not yet widely deployed. Upgrading to SNMPv3 is difficult and costly. However, it promises to provide for many GIG network management security requirements.

(U//FOUO) There are actually disadvantages of SNMPv2 versus SNMPv1 in that version 2 makes matters potentially worse from a security viewpoint. This is due to the fact that while both versions do not have security written into them, SNMPv2 introduces the concept of distributed management, which opens the management process to additional potential vulnerabilities. GIG implementations should only consider SNMPv3-compliant or equivalent systems.

2.5.3.3.2.1 (U) Implementation Issues

(U//FOUO) The addition of the security functions and their associated mechanisms to the SNMPv3 standard version has resulted in the fact that SNMPv3 is more compute-intensive than the earlier versions. This has led some in the research community to compare the efficiency of full SNMPv3 implementations with SNMPv2 running over Transport Layer Security TLS/TCP secure connections or, alternatively, over IPsec. These two options effectively separate out encryption protection from within the SNMP standard itself and bring it to a wrapping transport function. This only addresses the encryption/privacy aspects of SNMPv3 and does not implement any of the VACM access control functionality, which SNMPv3 provides us.

(U//FOUO) The Office of Naval Research (ONR) funded Midkiff and Hia of Virginia Tech in 2001 to look at the IPsec security option to SNMPv3 encryption across backbone networks. They showed that SNMPv3 could consume as much as 24% more network capacity than SNMPv2 over IPsec. The disadvantage of the IPsec method is that it does not provide for fine-grained access control. The advantage shown by the SNMPv2-over-IPsec solution was shown to deteriorate as the size of the application-layer payload increased. Much of the inefficiency of the SNMPv3 solution is due to the Basic Encoding Rules (BER) used to encode SNMP application data.

(U//FOUO) The NSA/ Laboratory for Telecommunications Science (LTS) funded Du and Shayman of the University of Maryland to investigate the performance comparisons of SNMPv1 over a TLS/TCP base with full SNMPv3 security. One issue of SNMPv1/TLS/TCP is the nontrivial overhead associated with setting up a session, as compared against SNMPv3 over UDP (sessionless). However, for a long session the costs of setting up the session are amortized over a large number of messages, and therefore the overhead per message decreases. The final experimental results showed that SNMPv3 (with full USM security functionality) session times were much larger (from 163% up to 433% of) than the comparable SNMPv1/TLS/TCP session times. Thus, for situations of lower data rate environments, this aspect of SNMPv3 may perhaps need to be considered.

2.5.3.3.2.2 (U) Advantages

(U//FOUO) SNMPv3 builds upon the general overall advantages of SNMP in that it solves many of the security problems of the earlier SNMPv1 and SNMPv2 versions. One of the basic appeals of SNMP has been its simplicity, because SNMP provides a bare-bones set of functions and thus is easy to implement, install, and use. If applied sensibly it won’t place an undue burden on the network. Moreover, due to its simplicity, interoperability can be achieved in a relatively straightforward manner—SNMP modules from various vendors can be made to work together with minimal effort.

2.5.3.3.2.3 (U) Risks/Threats/Attacks

(U//FOUO) The messages which will be needed to provide for assured GIG network management control and monitoring will be subject to a variety of potential adversarial threats or attacks. Hence, the security constructs of an enabling protocol such as SNMPv3 must be adequate to protect against these potential malicious actions. The SNMPv3 protocol's User-based Security Model (USM) improved upon the earlier versions of SNMP so as to protect against the following four threats:

- (U//FOUO) Modification of Information—Attempt by an unauthorized entity to alter an SNMP message in-transit (issued on behalf of an authorized principal)
- (U//FOUO) Masquerade—Attempt by an unauthorized entity to perform an operation by assuming the identity of an authorized entity
- (U//FOUO) Message Stream Modification—Delay or replay of messages to an extent greater than can occur in natural conditions of network service
- (U//FOUO) Disclosure—Attempt by an unauthorized entity to see the contents of SNMP message/data exchanges

(U//FOUO) SNMPv2 has been shown to be vulnerable to replay attacks (and resultant message stream modification) due to the possibility of clock time drift between network manager and remote agent. This is solved by SNMPv3—it supposedly would also be ameliorated by the adoption of a truly secure and robust Network Time Protocol (NTP) across the GIG. Though the SNMPv3 protocol provides for protection against the above 4 threats, it was decided during the development of SNMPv3 to not provide for defense against the following two threats:

- (U//FOUO) Traffic Analysis (TA)
- (U//FOUO) Denial of Service (DoS)

(U//FOUO) At the time of SNMPv3 definition it was deemed that these two threats either required defenses that were nearly impossible to achieve or were not as significant as the others.

(U//FOUO) While subject to various malicious threats or attacks—or merely to innocent network component failures—the GIG infrastructure will be subject to the potential risk that network management and control messages will be unable to reach their desired destinations. This is especially true in the case of an Internet IP protocol such as SNMP that provides all its signaling in-band (IB) on the same IP routing infrastructure upon which normal traffic travels. For example, in order to conduct management and control of a particular network router, the paths to that router will be necessarily operational or else the control function will not be possible. This quandary has led some industry proponents to propose that perhaps backup out-of-band (OOB), perhaps dial-up, control paths be maintained to at least the critical network elements.

(U//FOUO) While perhaps not as essential in the area of everyday network management and control, these OOB techniques may become most valuable during times of network fault monitoring and recovery. The possible segregation of SNMP traffic onto a physically separate management network would potentially require an entirely parallel architecture redesign (e.g., VLANs, routing, BGP/OSPF domains, new IP addresses, for configuring managers and remote agents). It would also require a transition plan to ensure continued management during migration. Carriers and other network service providers have used OOB for years because their businesses depend on the continuous availability of their network infrastructure. The degree to which the GIG should adopt this philosophy is yet to be determined.

(U//FOUO) The vulnerabilities of the original SNMPv1 protocol, with virtually no provision for security functionality, are such that many organizations purposely limit the use and application of SNMP. The newer SNMPv3, when and if fully deployed as specified, should go far to remove these concerns.

(U//FOUO) Meanwhile, however, the vulnerabilities of deployed SNMP systems continue to be exposed. An example of this is the work done in Finland during 2002 by the Oulu University Secure Programming Group (OUSPG). In this study more than four dozen vulnerabilities to SNMPv1 were demonstrated on commercial system implementations (e.g., Cisco). Examples of vulnerabilities include cases of seemingly innocent poor error handling when the SNMP primitive messages of Get, Set, or Trap were transmitted with invalid encodings or illegal internal values. The results of these simple non-malicious mistakes could lead to network elements crashing, locking up, rebooting, overwriting critical data values, or even enabling unauthorized access. Other uncovered vulnerabilities of SNMPv1 include the possibility of bounce attacks whereby malicious attackers could bounce their attacks off a trusted node.

(U//FOUO) Risks and vulnerabilities of SNMP have been well-documented by the US. Computer Emergency Readiness Team (CERT) and the CERT Coordination Center at Carnegie Mellon University's Software Engineering Institute (CMU SEI). Useful documentation available from them includes an SNMP Vulnerability FAQ (frequently asked questions—at http://www.cert.org/tech_tips/snmp_faq.html), which accompanies the illustrative "CERT Advisory CA-2002-03" on SNMP vulnerabilities (http://www.cert.org/advisories/CA-2002-03.html). CERT acknowledges the foundation work of OUSPG in the uncovering of many examples of vulnerable commercial SNMP deployed implementations.

(U//FOUO) Finally, even with the assumption of a finalized and robustly secure SNMPv3 standard, if the Request For Comments (RFC) are not fully and carefully implemented by the various vendors, there may still be residual vulnerabilities such as those to buffer overflow exploits. However, this can also be true of other network management standards.

2.5.3.3.3 (U) Maturity

(U//FOUO) SNMP has a fairly long history since its debut in the late 1980s. As such, it has had time to mature, certainly as proved by the development of the later versions through SNMPv3 in the late 1990s. This maturing process has been beneficial by solving many of the security issues left unresolved by the first version. The marketplace is populated by many implementations of SNMPv1, with marketplace adoption of SNMPv2 and SNMPv3 lagging due to business inertia reasons, while the standards process proceeds to improve upon SNMPv3. With the vulnerabilities of SNMPv1 having become well known, pressure will mount for retrofit with SNMPv3-compliant network management systems.

(U//FOUO) There are many commercial implementations of SNMP. These include systems built by HP, IBM, Novell, Sun, Microsoft, Compaq, Empire Technologies, Gordian, and SimpleSoft. In addition, there are at least 18 commercial or academic implementations of the more advanced SNMPv3, including those by AdventNet, BMC Software, Cisco, Halcyon, IBM, Multiport Corporation, SimpleSoft, SNMP Research, UC Davis, and University of Quebec. Thus, considering both the ongoing commercial work and the standards work within the IETF, SNMPv3 should continue to evolve and improve.

(U//FOUO) The various sub-technologies of the Integrity of Network Management/Control/Monitoring/Recovery technology area can be generally assigned Technology Readiness Level groups of Early, Emerging, and Mature.

- (U//FOUO) Basic SNMPv3 implementations—Mature (7 – 9)
- (U//FOUO) Key management enhancements for SNMPv3—Early (1 – 3)
- (U//FOUO) Efficient SNMPv3 with security by IPsec or SSL/TLS (rather than native SNMPv3 encryption)—Emerging (4 – 6)

2.5.3.3.4 (U) Standards

(U) U.S. Government Protection Profile on “Network Management” (http://niap.nist.gov/cc-scheme/pp/index.html).

(U//FOUO) As far as the definition of the SNMP protocols is concerned, there are a number of IETF RFCs that explain the relevant security-enabling aspects of SNMPv3. These include the following:

- (U) RFC 3414, “User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)”
- (U) RFC 3415, “View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)”

(U//FOUO) In addition to the IETF arena, a number of different standards groups have been developing competing or alternate frameworks for network management control and monitoring. These include the International Standards Organization (ISO) and the Open Software Foundation (OSF). However, these alternate approaches have for various reasons not yet been successful in the commercial marketplace. As such, reviewing these will be delayed until the upcoming "Alternatives" section.

2.5.3.3.5 (U) Cost/Limitations

(U//FOUO) There are several limitations currently to the broad implementation of SNMPv3. One of these is in the area of key management. The official SNMPv3 standard generically calls for initial OOB distribution of secret keys among manager and agent elements, without specifying a technique. Thus there is no accepted, standardized initial key distribution mechanism—only an experimental Diffie-Hellman approach. There is also no integration with centralized key management and authorization, such as RADIUS. One approach exists for Kerberos, but that has been labeled experimental, and Kerberos does not seem to be in wide commercial use. Finally, there has been only some initial work to standardize the widely desired Advanced Encryption Standard AES support (as described in a 2002 IETF draft, http://www.snmp.com/eso/draft-blumenthal-aes-usm-04.txt).

(U//FOUO) On a more positive note, however, very recent work during 2003-2004 has been undertaken on the SBSM (Session-Based Security Model) for SNMPv3. This would employ public key based I&A (between manager and agent elements), using the SIGMA key exchange protocol (using Diffie-Hellman). SIGMA has several advantages including its simplicity and efficiency, that it has had extensive review and is used for IKE (Internet Key Exchange), and it protects the identity of the session initiator.

(U//FOUO) The SBSM protocol itself has a number of advantageous characteristics and features:

- (U//FOUO) It uses existing security infrastructures for identity authentication
- (U//FOUO) Both ends of message exchanges are authenticated
- (U//FOUO) The responder agent reveals its identity and authenticates before the initiator manager
- (U//FOUO) Separate mechanisms are used for identity authentication as compared with message authentication or encryption,
- (U//FOUO) It has limited life time keys for encryption

(U//FOUO) The consequences of these features are that there is a low cost to creating new identities, changing, or deleting their authentication credentials. Also, saved encrypted messages can not be decrypted after an identity key is compromised. However, SBSM is a work in progress, and overall SNMPv3 key management will require some maturation and standards adoption.

2.5.3.3.6 (U) Dependencies

(U//FOUO) The future success of SNMP-based network management systems will depend upon their full adoption of SNMPv3 security functionality and the full marketplace adoption of SNMPv3 implementations in lieu of SNMPv1 systems. Finally, use of SNMP within the GIG will depend upon the demonstrated robust and correct implementations by vendors of SNMPv3, so as to minimize any residual vulnerabilities.

2.5.3.3.7 (U) Alternatives

(U//FOUO) Since SNMPv1 was originally proposed in the late 1980s, several competing standards alternatives have been proposed. Nonetheless, for a variety of reasons, SNMP continues to evolve and improve, whereas the competitors have often come and gone. SNMP-based network management and its associated security mechanisms continue to grow, expand its scope, and mature. Four examples of competing alternative architecture schemes are described below:

- (U) Common Management Information Protocol (CMIP) comes out of the ISO. The main problem with this protocol is that it is overly complex and perhaps overly ambitious. Due to this complexity, it can require up to 10 times the CPU power of an SNMP implementation. Few commercial implementations of CMIP can be found. CMIP originally was supposed to be the protocol that replaced SNMP in the late 1980s. It was funded by governments and large corporations, which caused many to believe that it would inevitably succeed. However, implementation problems delayed its widespread availability. CMIP had security advantages over SNMPv1 in that it included authentication and security log mechanisms. However, SNMPv3 solves the security holes of SNMP. Because of the fact that SNMP came out first and was much simpler to implement, CMIP is used today primarily in management of public telephone networks, while SNMP dominates most of the network management field.

- (U) Distributed Management Environment (DME) comes from the Open Software Foundation (OSF), originating during the 1991 timeframe (from proposals submitted by 25 organizations, including IBM, HP, Tivoli Systems, etc.). It is a framework meant for tackling the problem of managing distributed network devices. Unfortunately, it is not much used commercially. DME is an object-oriented environment (like CMIP). The main problem with DME is that it seems to over-generalize the framework. This causes a problem for the business interests of competing vendors (if SunNet Manager, HP OpenView, and IBM Netview all have the same GUI, protocols, etc., these platforms may lose bargaining position based on unique capabilities).

- (U) Hierarchical Network Management System (HNMS) comes out of the Network Attached Storage (NAS) domain. Its goal is to provide the capability to manage and monitor a very large Internet Protocol network. It relies on four types of modules: a server, a database, IO input/output modules, and UI user interface modules. All inter-module communication is done by the Hierarchical Network Management Protocol (HNMP). HNMP (like SNMP) is built on top of UDP/IP. Finally, four types of services are provided by HNMS:system parameters setting, data exchange, device discovery, and object management. In general, HNMS is more complex than SNMP and thus not as successful in the marketplace.

- (U) Hypermedia Management Architecture (HMMA) comes out of the Web-Based Enterprise Management (WBEM) initiative of the Distributed Management Task Force (DMTF) whose URL can be found at http://www.dmtf.org/standards/wbem/. It is a result of a movement to combine network management with system and desktop management. WBEM is supported by Microsoft, Compaq, Cisco, Intel, HP, etc. The idea is to integrate existing standards into a framework, combining Desktop Management Interface (DMI/RPC) for desktops/servers with SNMP for network management, and doing all related Internet communication through Hypertext Transfer Protocol (HTML/HTTP). This aggregated architecture can then be managed using any Web browser, which is an advantage over plain SNMP.

  (U) However, HMMA can be viewed not as a SNMP competitor but, rather as the long-awaited HTTP version of SNMP. The HMMP Protocol has been submitted to the IETF forum, and the HMMS Schema has been submitted to the DMTF forum. Of all the competitors to SNMP, HMMA perhaps has some chance of succeeding.

(U//FOUO) If a choice has been made to employ SNMP-based network management techniques, then an alternative to full SNMPv3 implementation would be to use non-native encryption (outside of the SNMPv3 specified techniques), such as IPsec or TCP/TLS (Transport Layer Security). This alternative encryption choice may prove to be more efficient in terms of computation burden, as compared with full SNMPv3 operation. Finally, as in the prior evaluations concerning out-of-band versus in-band network management, the ultimate alternative to in-band SNMPv3 would be to build a dedicated (physical) or dial-up backup network for network management purposes. And when it comes to the issue of fault management, consideration of HTTP over SSL has the problem of connection-orientation which would rule it out (as compared with SNMPv3).

2.5.3.3.8 (U) Complementary Techniques

(U//FOUO) As has already been shown, a complementary (or alternative) technique to the full implementation of SNMPv3 would be to implement SNMPv1 over IPsec or over TLS/TCP, due to the fact that SNMPv3 messages can require greater network capacity (mainly an issue only on lower data rate networks).

2.5.3.3.9 (U) References

(U) RFC 3414, “User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3),” http://www.ietf.org/rfc/rfc3414.txt , or http://www.faqs.org/rfcs/rfc3414.html, by Wijnen et al, December 2002.

(U) RFC 3415, “View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP),” http://www.ietf.org/rfc/rfc3415.txt , or http://www.faqs.org/rfcs/rfc3415.html, by Wijnen et al, December 2002.

(U) “TCM IA Architecture Overview” briefing, 30 June 2004, by NSA’s IAD TSAT IA IPT.

(U) "Understanding the SNMP Security Vulnerability," http://www.ins.com/downloads/seminars/SNMP_Vulnerabilities_13mar02.ppt, by Nicastro et al, March 2002.

(U) "Understanding the Risks of SNMP Vulnerabilities," http://www.lucent.com/livelink/255868_Whitepaper.pdf, by Davis et al, 2002.

(U) "SNMP's Real Vulnerability," http://www.networkmagazine.com/shared/printableArticle.jhtml;jsessionid=OHQF54CDNR3ISQSNDBCSKHQ?articleID=8703341, May 2002.

(U) "Researchers Reveal Major SNMP Holes," http://www.nwfusion.com/news/2002/0218snmp.html, February 2002.

(U) "Security in Network Management," http://opensores.thebunker.net/pub/mirrors/blackhat/presentations/bh-usa-97/Jeremy-snmp.ppt.

(U) "PROTOS Test-Suite: c06-snmpv1," http://www.ee.oulu.fi/research/ouspg/protos/testing/c06/snmpv1/, October 2002.

(U) "Security Comes to SNMP: The New SNMPv3 Proposed Internet Standards," http://www.cisco.com/warp/public/759/ipj_1-3/ipj_1-3_snmpv3.html, by William Stallings, The Internet Protocol Journal, December 1998.

(U) "CERT Advisory CA-2002-03 Multiple Vulnerabilities in Many Implementations of the Simple Network Management Protocol (SNMP)," http://www.cert.org/advisories/CA-2002-03.html, February 2002.

(U) "EE579T Network Security, 11: Firewalls and SNMP," Spring 2004 lecture by Dr. Richard Stanley of WPI Worcester Polytechnic Institute, http://ece.wpi.edu/courses/ee579t/EE579T-Class%2011C.ppt.

(U) "The AES Cipher Algorithm in the SNMP's User-based Security Model," IETF Internet Draft by Blumenthal et al, http://www.snmp.com/eso/draft-blumenthal-aes-usm-04.txt, October 2002.

(U) "Implementation and Performance Analysis of SNMP on a TLS/TCP Base," by Du et al, http://www.umiacs.umd.edu/docs/Du.ppt.

(U) "Securing SNMP Across Backbone Networks," by Hia et al, http://fiddle.visc.vt.edu/courses/ece5566/lectures/21securesnmp.pdf, 2001.

(U) "Deploying Secure SNMP in Low Data Rate Networks," by Hia et al, http://www.irean.vt.edu/navciiti/reports/secure_snmp_nov_2000.pdf.

(U) "Session-based Security Model for SNMPv3 (SNMPv3/SBSM)," http://net-snmp.sourceforge.net/sbsm/SBSM.ppt.

(U) http://www.nanog.org/mtg-0405/pdf/hardaker.pdf

(U) http://www.net-snmp.org/sbsm/SBSM-bof-wes.ppt

(U) http://www.net-snmp.org/sbsm/

(U) IETF drafts during 2004:

- (U) http://www.net-snmp.org/sbsm/draft-perkins-snmpv3-overview-00.txt
- (U) http://www.net-snmp.org/sbsm/draft-hardaker-snmp-session-sm-01.txt

(U) "SNMP Architecture Alternatives," (U) http://www2.rad.com/networks/1999/snmp/index.htm.

(U) "SNMP Vulnerabilities Frequently Asked Questions (FAQ)," (U) http://www.cert.org/tech_tips/snmp_faq.html.

2.5.4 (U) Assured Resource Allocation: Gap Analysis

(U) Gap analysis for the Assured Resource Allocation Enabler indicates that the main areas of future development are as follows:

- (U//FOUO) Need to develop SAR (Security Aware ad-hoc Routing) protocol capability that will work in tactical wireless GIG contexts.

- (U//FOUO) More generally, need to verify that flexible and security-cognizant routing protocols such as FIRE (Flexible Intra-AS Routing Environment) can be implemented across the GIG and that the needed security QoS parameters (and associated routing table information) can be passed to GIG routers across any intervening red/black boundaries.

- (U//FOUO Need to develop a GIG Quality of Protection standard that will be a foundational element of the IA Policy-based Routing capability.

- (U//FOUO) Need to develop a robust MLPP precedence and preemption standard for the GIG that will be well-integrated with the required foundation of a GIG PMI Privilege Management Infrastructure. Operational-based resource allocation/deallocation actions will demand that the associated privileges be consistently valid and universally distributed to needed policy enforcement points.

- (U//FOUO) Need to flesh out the capabilities of SNMPv3, if this protocol is decided as the way to go for network signaling security (as this document suggests). SNMPv3 is fairly mature, except for key management aspects. Also need to validate that SNMPv3 will be efficient enough when widely applied throughout the GIG.

- (U//FOUO) Need to develop a Protection Profile for Network Management.

(U//FOUO) Technology adequacy is a means of evaluating the technologies as they currently stand. This data can be used as a gap assessment between a technology's current maturity and the maturity needed for successful inclusion in the GIG.

(U//FOUO) Table 2.5-1 lists the adequacy of the Assured Resource Allocation technologies with respect to the IA attributes discussed in the RCD.

Table 2.5-1: (U) Technology Adequacy for Assured Resource Allocation

This Table is (U)

		Technology Category			
		IA Policy-based Routing	**Operational-based Resource Allocation**	**Integrity of Network Management /Control /Monitoring /Recovery**	**Required Capability (attribute from RCD)**
Enabler Attribute	**Standard**				IAAV1-IAAV4, IACNF6, IANMA2, IANMP1-IANMP5, IAAV21, IARC01 – IARC12, IAMP02
	Secure Solution				IACNF6, IACNF12, IANMA3, IANMP4, IANMP5, IARC01 – IARC12
	Scalable Solution				IAAV1-IAAV4, IAAV15, IAFM1, IANMA2
	Protection Profile	N/A	N/A		
	High Assurance				IACNF6, IAFM1, IANMP1-IANMP5, IAAV20
	Distributed/ Global Reach				IAAV1-IAAV4, IAAV15, IAFM1, IAFM3, IAFM4, IANMA3, IANMP1-IANMP5
	Verifiable Solution				IAAV15

This Table is (U)

(U//FOUO) In summary, the SNMPv3 standard is fairly mature (accounting for the black cell in the matrix). At the current time, there is only provision for a Protection Profile dedicated to Network Management. It is noted that there is a Protection Profile for "Switches and Routers" in general (see http://niap.nist.gov/cc-scheme/pp/index.html). It is generally viewed that technology for operational-based resource allocation is less mature and therefore less adequate for the GIG than the available IA-based routing and network management technologies. Various sub-technologies are available for the latter two areas but need to be integrated together and operationally validated.

2.5.5 (U) Assured Resource Allocation: Recommendations and Technology Timelines

(U) The following is a list of recommendations for advancing the technologies required for the successful implementation of this GIG enabler:

- (U//FOUO) Encourage the further development of adaptive security-driven (i.e., IA policy-based), wireless routing algorithms (such as SAR) for inclusion in JTRS and WIN-T
- (U//FOUO) Advance the standards evolution and demonstration/implementation of extensible routing protocols (such as OSPF and IS-IS) so that IA metrics can be fully employed in routing decisions
- U//FOUO) Encourage the development of a GIG Precedence and Preemption standard that is closely tied with the required corollary GIG Privilege Management Infrastructure. The overall GIG Precedence/Preemption standard should ideally include the new GIG-BE-based VoIP-evolved DISN MLPP protocol as a subset capability
- (U//FOUO) Advance, as an inclusion to the GIG Precedence and Preemption standard, the capability for rational post-preemption rescheduling so as to not leave GIG customers without requested services
- (U//FOUO) Support developments that will ensure that an operational-based resource allocation infrastructure will have GIG-wide (i.e., worldwide) reach in its customer adjudication process (especially in the case of multiple requests and possible GIG congestion)
- (U//FOUO) Push for the development of effective and scalable key management mechanisms for SNMPv3 messaging
- (U//FOUO) Continue to follow the efficiency issue/impact of SNMPv3 native encryption (as being about 20%+ slower than SNMPv2 over IPsec or SSL)
- (U//FOUO) Continue to track potential competing technologies/standards to SNMPv3 for network management/control/monitoring (even though various competitors have come and gone)

(U//FOUO) Figure 2.5-11 contains preliminary technology timelines for this IA System Enabler. These are the result of research completed to date on these technologies. As the Reference Capability Document and the research of technologies related to these capabilities continues, these timelines are expected to evolve. The timelines reflect when the technologies could be available—given an optimum set of conditions (e.g., commercial community evolution starts immediately, GOTS funding is obtained, staffing is available). Technology topics with missing timelines (if any) indicate areas where further work is needed to identify the milestones.

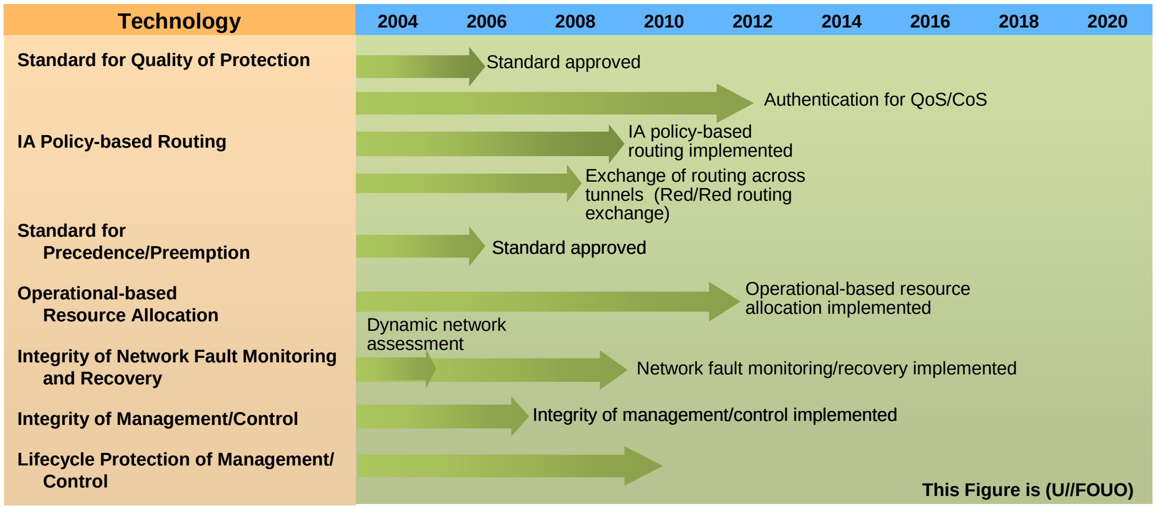


Figure 2.5-11: (U) Technology Timeline for Assured Resource Allocation

2.6 (U) NETWORK DEFENSE AND SITUATIONAL AWARENESS

(U//FOUO) Network Defense and Situational Awareness is the IA System Enabler that allows the GIG to achieve the IA Mission Concept of Defend the GIG. It consists of enterprise-wide protection, monitoring, detection, and analysis that provide input into situational awareness of the operational mission(s) being carried out and result in response actions. The collection and analysis of sensor information—coupled with intelligence data, operational priorities, and other inputs—enables the creation of user-defined operational pictures of the assurance of GIG resources. The analysis of this information supports the development of situational awareness and the identification and characterization of potentially hostile activity.

(U//FOUO) Situational awareness enables an adaptive and rapid adjustment of enterprise resources in response to unauthorized network activity and sub-optimal GIG resource configurations to support and achieve the six GIG IA Mission Concepts.

(U//FOUO) This IA System Enabler consists of the following major functions as defined in the Joint Concept of Operations for Global Information Grid NetOps, 20 April 2004:

- (U//FOUO) Protection: Prior actions taken to counter vulnerabilities in GIG information transport, processing, storage, service providers, and operational uses. Protection activities include emission security (EMSEC), communications security (COMSEC), computer security (COMPUSEC), and information security (INFOSEC)—all incorporating access control, cryptography, network guards, and firewall systems

- (U//FOUO) Monitor: The monitoring of information systems to sense and assess abnormalities, the use of anomaly and intrusion detection systems. (Monitoring also includes receiving input from network monitoring as well as from a wide variety of real-time and status reporting)

- (U//FOUO) Detection: Timely detection, identification, and location of abnormalities—to include attack, damage, or unauthorized modification—is key to initiating system response and restoration actions. [Detection also includes actions taken in anticipation of an attack (i.e., configuration adjustment)]

- (U//FOUO) Analyze: Assessing pertinent information to [achieve] situational awareness, evaluate system status, identifying root cause, defining courses of action, prioritizing response and recovery actions, and conducting necessary reconfiguration of GIG assets as needed

- (U//FOUO) Response: Directed actions taken to mitigate the operational impact of an attack, damage, or other incapacitation of an information system. Response also includes restoration—the prioritized return of essential information systems, elements of systems, or services to pre-event capability. (Coupled with restoration is the ability to undo a response)

2.6.1 (U) GIG Benefits due to Network Defense and Situational Awareness

(U//FOUO) The Network Defense and Situational Awareness System Enabler provides the following benefits to the GIG:

- (U//FOUO) Dynamic protection of GIG network and computing resources from attack (attack being defined here as a sequence of one or more exploits or other actions taken by an adversary that lead to success of the adversary's mission); updated defensive posture based on near-real-time detection, intelligence, and operational and network information to enable a rapid response
- (U//FOUO) Continuous, assured (e.g., availability, confidentiality, integrity) discovery, collection, processing, correlation, storage, and dissemination of intrusion detection and audit data. IA services applied to the sensor and audit resources ensure the availability, integrity, and confidentiality of the information received and also enable the authentication of the source
- (U//FOUO) Detection and sharing of events and anomalies at multiple tiers (i.e., local, regional, global) within the GIG. User-defined operational pictures (UDOP) of the situational awareness information will enable analysis at all tiers and response to events as they occur
- (U//FOUO) Trusted, real-time, user-defined operational picture of the IA/security posture of the GIG at any tier. Building upon the assured discovery, collection, processing/analysis, storage and dissemination of intrusion detection information, and audit and network management data, authorized users will be able to customize their view into the GIG as required to meet operational needs and also continuously monitor GIG network activity
- (U//FOUO) Rapid analysis and response alternatives developed and modeled. Collection of sensor information, audit data, and network management data is only one step in the process. Being able to rapidly analyze that information requires greatly enhanced correlation, analysis, and modeling tools over what is currently available today in order to determine if an attack is occurring or imminent, and what the impact of such an attack might be if not countered
- (U//FOUO) Enterprise-wide tools will enable the capability to rapidly monitor, analyze, and respond to system, computing, and network attacks, degradations, outages, misuse of resources, and events such as changes in operational priorities
- (U//FOUO) Automatic and global intelligent (self-learning) defensive action enforcement to contain, recover, restore and undo, and reconstitute the GIG. Having determined an attack is underway—or imminent—and with likely resulting damage, alternative defensive countermeasures can be postulated and modeled/evaluated before implementation throughout the GIG

- (U//FOUO) Governance of response actions. There are potential legal ramifications to employing defensive countermeasures to an attack. The analysis and modeling that will be available will strengthen the legal position that all due diligence was taken to analyze alternatives before deploying any response

- (U//FOUO) Automatic prediction of attack strategies, objectives, and targets based on intrusion detection data, network data, and attack patterns. Automated tools performing trend analysis of sensor data and log files will provide the GIG with the capability to predict when and where identified attacks may appear elsewhere on the network

2.6.2 (U) Network Defense and Situational Awareness: Description

(U//FOUO) Network Defense and Situational Awareness is a critical enabler to provide the protection and support needed to achieve the GIG Mission Concept of Defend the GIG. This enabler defines actions taken to protect against, monitor, detect, analyze, and respond to potential and actual unauthorized network activities as well as unintentional non-malicious user error that could potentially harm the GIG. A concerted effort is required to find solutions to current technology issues related to an accurate view of organic system strengths and weaknesses for an enterprise of information on the scale of DoD’s to be secure, available, and responsive to operational requirements.

(U//FOUO) As a measure of effectiveness, DoD-wide system administration is highly dependent upon an accurate, real-time understanding of the configuration and situational awareness of DoD networks. Adversaries may periodically identify a weakness in a system, exploit that weakness, and then return the system to its original state. In addition, multiple attacks and exploitations can occur simultaneously and affect multiple missions. The planning of appropriate Courses of Action (COAs) will require constant awareness of the system and network configuration/state, which can lead to an overwhelming amount of data that needs to be analyzed. As a result, the task for administrators and analysts to understand how disparate attacks on a network affect an ongoing mission(s) and subsequently determine effective countermeasures becomes even more difficult.

(U//FOUO) Distributed sharing of information is an important capability and begins with the monitoring and collecting of sensor information across the GIG. Referring to Figure 2.6-1, it can be seen that sensor information will be gathered from various locations and at all levels to include local (Tier 1), regional (Tier 2), and global (Tier 3) tiers. Information will be shared across all tiers, to include both peer-to-peer but also vertically within the organization. Further, while there might be a loss of information as it traverses horizontally and vertically, it is critical to have the ability for higher functions in the vertical space to be able to drill-down into specifics of a lower tier's data.

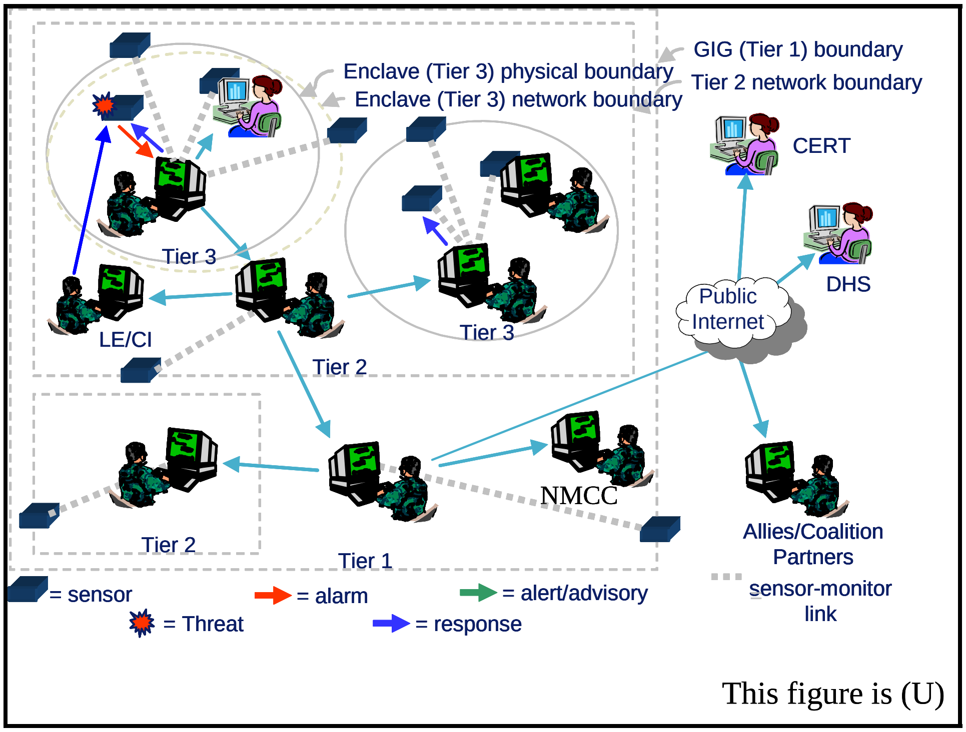


Figure 2.6-1: (U) Representative Sensor Configuration

(U//FOUO) Today, sensors are primarily distinct special purpose devices (e.g., Intrusion Detection Systems [IDSs], Intrusion Prevention Systems [IPSs], Firewalls, Guards)—providing the information that is monitored and analyzed. In the future, every node and CND device on the network will provide sensor information from its unique perspective and that will be coupled with intelligence information, mission priorities, and audit logs to create a much broader view of the operational picture. Sensors can be grouped in zones that are defined by geography, function, and security. Zone/node sensors that can operate on the concept of reporting status changes to their nearest neighbor will also be integrated into the GIG.

(U//FOUO) A major goal of the GIG is to provide a Black Core for the data sent across it to transit. The term Black in this sense means that the data traversing the GIG is encrypted, and if necessary, also integrity-protected. Performance/situational monitoring and analysis of mixed mode Black Core will require a change to sensor strategy. Sensors that require access to encrypted information will need to be located before encryption. This introduces a host of new challenges, including management and control of distributed sensors and sensor collection and processing across multiple classification boundaries.

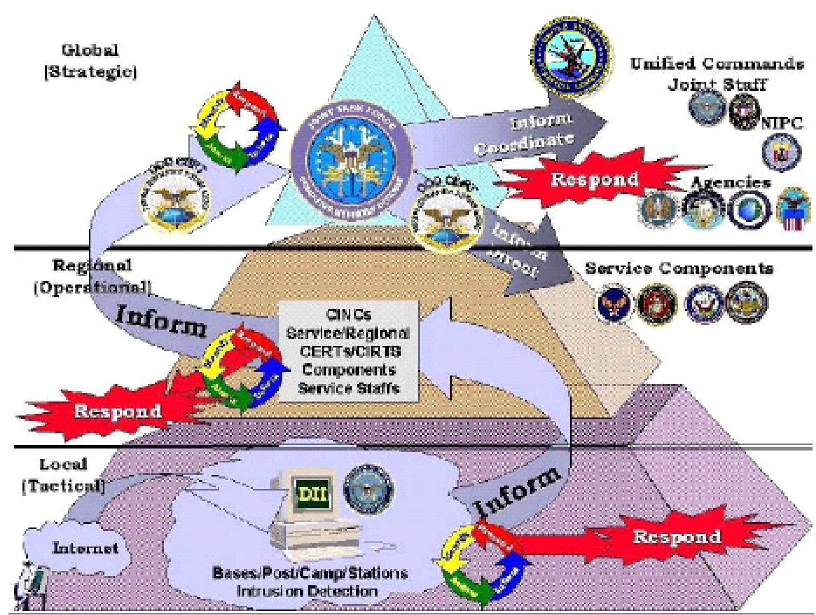
(U//FOUO) While this notion of a Black Core provides significant confidentiality and data integrity protection, it can also limit the ability of the GIG core itself to detect attacks. First, if all data packets are encrypted at the IP level (e.g., by HAIPEs or commercial IPsec implementations), the GIG cannot detect the contents of the packets, and thus cannot detect viruses, worms, or other malicious logic. As a result, the source of an attack may be hidden. Information from the red IP header will need to be made available to the black IP header.

(U//FOUO) Similarly, if the IP traffic is being tunneled (that is, there is a black IP header wrapped around the actual traffic), the GIG core may not even be able to tell where data packets are originating. At best, the GIG core can only tell that there is an unusual amount of traffic (e.g., either much larger amounts of traffic than is normal or a usually-busy link goes quiescent). The GIG cannot directly tell that an attack is under way; nor can it launch a response to that attack. In this case, the only place where attacks can be detected, and the only place from which a response can be effected, is the application-layer code at the end system.

(U//FOUO) Based on the size and complexity of the GIG, CND capabilities will need to be available for high volume, high speed connections to a variety of services (i.e., provider services, coalition services, and cross-domain services). Monitoring and collection of sensor information from coalition users and devices connected to the GIG is a serious concern.

(U//FOUO) A non-DoD entity interface specification is needed to identify what minimum sensor information is required and how it is to be provided. This specification must also address how defensive actions will be promulgated to coalition partners. Correlating sensor information received from various networks will introduce additional challenges.

(U//FOUO) Detection of anomalous behavior, detection of attack, quality of service, deviations from expected communication patterns, and all sorts of detailed monitoring provide the capability to ensure the integrity of individual GIG services and the enterprise-wide assurance of all managed information systems. Referring to Figure 2.6-2, it can be seen that if anomalous or attack activity is detected then the appropriate response will be performed at each tier.



This figure is (U)

Figure 2.6-2: (U) Representative Flow of Situational Awareness Data

(U//FOUO) In addition, information is passed to the next higher tier for activities requiring a wider view, analysis, and response. An assumption can be made that anomalies and attacks will be detected at the lowest tiers. While this may be true today, more sophisticated attacks and anomalies can only be detected if a wider view is obtained from the outset. Consequently, while real-time data passed to higher tiers consists of only that which is necessary for real-time response, detailed data should also be made available periodically for off-line analysis to identify trends and to apply algorithms for low-intensity attacks, intrusions, and exploitation.

(U//FOUO) Authorized users must be guaranteed ready access to all information contributing to situational awareness. ad Authorized users also must be able to verify the integrity and source of origin of the information—and in some cases ensure the confidentiality of the information. To do this, many different IA capabilities must be enabled. To determine that a user is authorized requires that there be mechanisms to provide assured identities to all users and mechanisms to derive an authenticated session score to confirm that a user identity has been authenticated to some level of assurance. A digital access policy will specify how access control to sensor information and any operational displays will be enforced and under what conditions exceptions to the policy will be managed. Management of sensor resources will fall under Section 3.5, the Assured Resource Allocation Enabler.

(U//FOUO) Managers and users of the GIG need near real-time awareness of current threats, configuration, status, and performance of the GIG and its components. A trusted UDOP is a tailored view of an operational cyberspace picture. The GIG will provide relevant situational views of GIG operations at any level, with aggregation and event correlation to the higher levels and from peer-to-peer. Automated situational views will be enabled through:

- (U//FOUO) Continuous monitoring of GIG configuration, status, and performance
- (U//FOUO) Posting of situational awareness information (raw and processed)
- (U//FOUO) Assembly of situational awareness information (monitored data plus threat and operational priorities)
- (U//FOUO) Storage of situational awareness information. In addition to intrusion detection information, situational awareness will encompass network management data, intelligence findings, operational missions, operational mission requirements and priorities, and IA service status

(U//FOUO) The CND component of the GIG will also provide the capability to take appropriate action on processed situational awareness data:

- (U//FOUO) Automated display modifiable to suit each level of GIG management
- (U//FOUO) Enterprise-wide mapping of services/applications to identify and mitigate vulnerabilities of all DoD hosts and associated services and applications
- (U//FOUO) Enterprise-wide tools to rapidly evaluate, analyze, and respond to system and network attacks, degradations, outages, and events

- (U//FOUO) Ability to rapidly adjust the GIG configuration based on different cyber Information Operations Condition (INFOCON) levels to best respond to identified situations

(U//FOUO) The GIG will have the capability to immediately identify, detect, and respond appropriately to anomalies, attacks, or disruptions from external threats, internal threats, and natural causes. Once the event has occurred, the GIG will have the capability to implement mission impact analysis/battle damage assessment. The GIG will have the automated response capability to globally enforce intelligent (self-learning) defensive actions that contain, recover, restore, and reconstitute the GIG (e.g., automatically block DoS attacks traffic to vulnerable DoD hosts, and counter attack). Response actions will be coordinated across a broad range of operational elements, including Enterprise Service Management for configuration management and restoration of disrupted or degraded capabilities.

(U//FOUO) Cyber attack attribution will play an essential role in identifying attackers and deterring further attacks. These capabilities will provide attacker/attack profiles and fingerprinting, trace to true country of origin, as well as provide complete trace-back and geolocation attackers. Forensic data will be captured and shared with Law Enforcement and Counter-Intelligence to investigate and if warranted prosecute perpetrators of unauthorized activities.

(U//FOUO) As a complementary mechanism, a network capability will collect and assess network data to provide warnings of compromise to CND command and control elements, and information will be further disseminated to subordinate CND organizations. It will provide CND analysis of network data to detect if a severe compromise calls into question the integrity of the GIG.

2.6.3 (U) Network Defense and Situational Awareness: Technologies

(U//FOUO) The following technology areas support the Network Defense and Situational Awareness IA System Enabler:

(U) Note: For convenience of analysis and organization, the technologies have been grouped together by the major function it is most designed to effect. This is not meant to suggest that the following technologies can only support one function, as many span multiple functions.

(U) Protection

- (U) Protect Technologies
  - (U) Firewalls
  - (U) Filters/Guards
  - (U) Anti-Virus, Anti-SPAM
  - (U) Disk and File Encryption
- (U) Deception Technologies
  - (U) Honeypot
  - (U) Honeynet

(U) Monitor

- (U) Situational Awareness
  - (U) User-Defined Operational Picture (UDOP)
  - (U) Network Operations (NETOPS)
- (U) Network Mapping
  - (U) Vulnerability Scanning

(U) Detection

- (U) Intrusion Detection Systems (IDS)
  - (U) Host-Based IDS, Network-Based IDS
  - (U) Misuse Detection, Anomaly Detection
- (U) Intrusion Prevention Systems (IPS)
  - (U) Host-Based IPS, Network-Based IPS
- (U) User Activity Profiling

(U) Analyze

- (U) Cyber Attack Attribution
  - (U) Traceback
- (U) Correlation Technologies

(U) Response

- (U) CND Response Actions
  - (U) Courses of Action (COAs)
- (U) Automated IA Vulnerability Alert (IAVA) Patch Management

2.6.3.1 (U) Protect Technologies

2.6.3.1.1 (U) Technical Detail

(U) The ability to protect GIG network assets from computer network attack is a key cornerstone of computer network defense (CND) capabilities. A robust CND architecture includes both defense-in-depth and defense-in-breadth:

- (U) Defense-in-depth - multiple layers of protection through the network against a particular attack type
- (U) Defense-in-breadth - protection against various attack types through and across the network

(U) Protection capabilities tend to be the first line of defense against network attacks as well as the propagation of potentially harmful non-malicious user activity. Less sophisticated adversaries can often be deterred by the sheer existence of protect technologies in today's network architectures. The most straightforward example of this is the placement of stateful firewalls at network perimeters that serve to deflect automated scanning and probing activity.

(U) Current protect technologies are for the most part limited to static defenses against known attack types. They include, but are not limited to, the following technology areas:

- (U) Network-Based Firewalls - The most common current implementation of protect technologies is network firewalls situated at perimeter boundaries to restrict data communications to and from one of the connected networks [RFC 2828]. These firewalls often provide the division between intranets and the Internet and come in both stateful and non-stateful varieties
- (U) Host-Based Firewalls - Includes software application firewalls such as those that come pre-packaged with operating systems as well as independent commercial software firewalls, and hardware-based firewalls resident on the network interface card. Current hardware-based firewalls are highly resistant to attacks that successfully gain user access to a host

- (U) Network Filtering Devices - A means of restricting data communications between connected networks—often implemented on network routers. These filtering devices can act as primitive non-stateful firewall devices
- (U) Application Filters - A means of restricting data communications at the application layer (e.g., wrappers)
- (U) Virus Protection - Software designed to search hard drives and disks for known viruses and then quarantine any found
- (U) Disk and File Encryption - Software designed to encrypt portions of a disk to protect data while not in use
- (U) Guards - Guards are generally used to prevent unauthorized data transfer between security domains. Hence, guard technology is discussed in Section 2.3.

2.6.3.1.2 (U) Usage Considerations

2.6.3.1.2.1 (U) Implementation Issues

(U//FOUO) Many protection mechanisms are currently implemented and managed on a device-by-device or application-by-application basis. This presents significant challenges in a distributed advanced system such as the GIG where implementation and management is designed with a tiered approach for all levels. For a network of this scale, it will be necessary to deploy technologies with advanced, centralized management capabilities.

(U//FOUO) The current trend in patch management also presents significant issues within the GIG. Many commercial operating systems and applications, including virus protection software, rely heavily on regular updates and patches to maintain up-to-date protection capabilities. This approach is rudimentary at best, since it requires secure web portals, accurate and trusted update code without inadvertent consequences, and valuable bandwidth.

2.6.3.1.2.2 (U) Advantages

(U) There is a clear advantage to preventing malicious activity before it reaches its intended target. Preventing an attack is far more desirable than detecting an attack—then responding to and recovering from it. The better we do the former, the easier it will be to do the latter. We cannot assume, however, that all attacks can be prevented, and therefore we must rely on a full breadth of CND capabilities to defend the network.

(U) Network-based protection systems such as perimeter firewalls and network filtering devices offer the advantage of protecting entire enclaves from many types of attack at the gateway between the Internet and an intranet.

(U) Host-based protection systems, on the other hand, push the protection capabilities to the network endpoints. Adversaries frequently consider these endpoints, often user workstations, to be attractive and more vulnerable targets to attack. By placing resilient host-based firewalls on the individual workstations, the defensive posture is increased significantly and makes them less-attractive targets. An additional advantage is that even if one workstation is compromised, the adversary still does not have open access to other workstations on the intranet. By pairing host-based firewalls with network-based perimeter firewalls, an additional layer is added to the defense-in-depth architecture.

(U) The application filters technology area, including virus protection, provides the advantage of another defense-in-depth layer against cyber attack, this time at the application layer. A variety of wrappers have been developed to intercept system calls intended to exploit an application, operating system, or host access. Commercial filters exist to scan email for malicious attachments. This approach to protect workstations can stop attacks such as worms from propagating past the infected host or further infecting the host.

(U) Disk and file encryption, a current COTS technology area, provides the advantage of encrypting file data stored on hard drives. This increases the work factor required by the adversary to access the file (the higher the number of encryption bits the longer it will take to crack).

2.6.3.1.2.3 (U) Risks/Threats/Attacks

(U//FOUO) Details of the GIG IA Risk Assessment, including detailed risks, threats, and attacks, are provided under separate cover. A fair amount is known about today’s adversaries, and their goals and techniques. Unfortunately, very little can be said about the 2020 adversaries; thus making protecting against them a significant challenge.

(U//FOUO) Results of the risk assessment indicate that protect technologies can in some cases provide a control surface for the adversary to launch an attack against the GIG. This is a risk that must be carefully considered when both designing and integrating protection mechanisms. The CND architecture must be designed so that protect technologies do not introduce vulnerable choke points. One approach to addressing this is to push the protection capabilities to the end point workstations rather than at network perimeters. Another approach in use today is redundancy in the architecture.

2.6.3.1.3 (U) Maturity

(U) There is much room for improvement in protect technologies.

- (U) Current technologies are vulnerable to network attack and must be designed for robustness
- (U) Protection must be designed throughout the network, not just at the perimeters. Adversaries often target the weaker, less protected network endpoints such as workstations
- (U) Protection must be designed into all network components, not band-aids placed over weak Commercial-off-the-Shelf (COTS) and Government-off-the-Shelf (GOTS) devices

- (U) Must be designed to be effective in encrypted network environments
- (U) Must be able to prevent attacks as close to the attack source as possible. This requires the ability to first detect where the source is at the onset of the attack
- (U) Must do a better job of protecting against an adversary with insider network access

(U//FOUO) Because COTS products are widely available and have been so for years, protect technology is rated as Mature (TRL 7-9).

2.6.3.1.4 (U) Standards

(U) There are no current standards for protect technologies. Any standards should be closely tied to those for intrusion detection as a whole, in particular if the protect technology reports unusual behavior to a centralized monitoring or analysis engine.

(U) The following Protection Profiles have been evaluated and certified with NIAP (http://niap.nist.gov/cc-scheme/pp/index.html):

- (U) U.S. Government Firewall Protection Profile for Medium Robustness Environments, Version 1.0
- (U) Application-Level Firewall for Basic Robustness Environments Protection Profile, Version 1.0
- (U) Application-Level Firewall for Medium Robustness Environments Protection Profile, Version 1.0
- (U) Traffic Filter Firewall Protection Profile for Medium Robustness Environments, Version 1.4
- (U) Traffic Filter Firewall Protection Profile for Low Risk Environments, Version 1.1

2.6.3.1.5 (U) Cost/Limitations

(U) Protect technologies range from inexpensive, such as host-based virus filters, to moderately expensive, such as perimeter firewalls.

(U) The requirement to continuously update today's protect technologies with security patches and new signature downloads is a significant limitation to their usefulness and survivability. Current industry practice is to issue a constant stream of patches that must be evaluated and implemented—requiring significant management overhead and annual licensing agreements. Without the most recent updates, these systems remain vulnerable to a variety of attacks, many of which are readily downloaded from the Internet.

(U) A disadvantage of network-based protection systems is that once an attack pierces the edge device, it can cause widespread harm within the intranet. This is especially the case if little or no internal protection systems are in place.

(U) A disadvantage of host-based protection systems is that a greater number of protection systems must be managed. Centralized management capabilities will be critical to this architectural approach.

(U) The disadvantages of application filters include scalability with technologies that do not have centralized management systems, complexity associated with customization per user behavior, and any reliance upon signatures that must be updated on a regular basis.

(U) The disadvantage of disk and file encryption is that when a file or encrypted partition is being accessed, it is decrypted and vulnerable. This is an inexpensive technology available today.

2.6.3.1.6 (U) Dependencies

(U) The ability to adequately protect a network relies heavily on maintaining control of the GIG assets as well as enforcing strong policies and procedures which GIG users are bound to follow.

2.6.3.1.7 (U) Alternatives

(U) The alternative to wide deployment of protect technologies is the incorporation of a strong IA architecture within the GIG.

2.6.3.1.8 (U) Complementary Techniques

(U) A resilient GIG network with a strong IA architecture goes a long way to provide protection against cyber attack and can therefore be considered a complementary technique. For example, encrypted network segments, use of strong authentication, and well written software immune from buffer overflow attacks can all serve to prevent network attacks. There will be holes, however, that protect technologies will serve to plug.

2.6.3.1.9 (U) References

(U) “A Public Key Infrastructure for the Secure Border Gateway Protocol (S-BGP),” by K. Seo, C. Lynn, and S. Kent, DARPA Information Survivability Conference and Exposition II, Volume 1, June 2001, pp. 239-253.

(U) “Document Integrity through Mediated Interfaces,” by M. Tallis and R. Balzer, DARPA Information Survivability Conference and Exposition II, Volume I1, June 2001, pp. 263-272.

(U) “Dynamic VPN Communities: Implementation and Experience,” by D. Kindred and D. Sterne, DARPA Information Survivability Conference and Exposition II, Volume 1, June 2001, pp. 254-263.

(U) Internet Security Glossary, Version 2, 20 August 2004 [replaces RFC2828].

(U) “Preventing Denial of Service Attacks on Quality of Service,” by E. Fulp, Z. Fu, D. Reeves, S. Wu, and X. Zhang, DARPA Information Survivability Conference and Exposition II, Volume I1, June 2001, pp. 159-174.

(U) “Security at the Network Edge: A Distributed Firewall Architecture,” by T. Markham, and C. Payne, DARPA Information Survivability Conference and Exposition II, Volume 1, June 2001, pp. 279-286.

(U) http://www.securecomputing.com/

(U) http://www.symantec.com

2.6.3.2 (U) Deception Technologies

2.6.3.2.1 (U) Technical Detail

(U) Information systems have seen a growth in size and complexity over the past several years. Unfortunately, the ability to defend these systems has not evolved as quickly as the growth in the sophistication, tools, and techniques of attackers. Attackers are constantly developing new avenues for exploitation. Fortunately, research activities over the past several years have produced new technologies that will support a more advanced and layered approach to security.

2.6.3.2.1.1 (U) Honeypots

> (U) "A honeypot is an information system resource whose value lies in unauthorized or illicit use of that resource." - Lance Spitzner

(U) Honeypots, also known as deception-based mechanisms or decoy-based intrusion protection, are specifically designed to attract an attacker's attention away from an operational system into an environment where the attacker can be observed and monitored—ideally without the attacker's knowledge.

(U) The intention of honeypots is not to capture an attacker or to thwart an attack, but rather to allow an attack to proceed in a controlled manner as a means to monitor and gather information about new techniques and methods used to compromise systems. This must be done while carefully balancing the benefit of learning the attacker's methods against the risk that a compromised system will be used as a launching point to attack real operational systems or other systems on the network.

(U) In general, there are two ways that honeypots are implemented:

- (U) Production – primarily used by companies or corporations to protect against an attack, easy to use, but capture limited amounts of information

- (U) Research – primarily used by research, military, or government organizations, complex to deploy and maintain, but capture extensive amounts of information

(U) The two general types of honeypots are:

- (U) Low-Interaction Honeypots – requires less monitoring, limited interaction, normally work by emulating services and operating systems

- (U) High-Interaction Honeypots – requires more monitoring, more complex, normally involve real operating systems and applications

(U) Low-Interaction Honeypots - Low-interaction honeypots, such as Honeyd, work on the concept of monitoring unused IP space. Once an attack is attempted, the connection is intercepted and redirected to an emulated service. The honeypot is then able to detect and log the activity, as well as capture all of the attacker's interaction with the emulated service. In some honeypots, actual operating systems can also be emulated.

(U) High-Interaction Honeypots - High-interaction honeypots, such as honeynets, offer an attacker an entire network of computers that are designed to be attacked. Within this highly controlled network, nothing is emulated or assumed. The idea here is to allow the attacker to find, attack, and break into these systems while controlling and capturing every activity.

2.6.3.2.1.2 (U) Honeynets

(U) As previously discussed, honeypots are deception devices within an operational network to learn an attacker's behavior and techniques. Honeynets, on the other hand, are an entire network of deception devices and are considered a combination of high-interaction honeypots. Their purpose is not focused on a specific operational environment, but rather to research an attacker's behavior in general. Also, honeynets are excellent tools for learning how to set up and manage all aspects of operational systems including traffic analysis, intrusion detection systems, system log and audit capabilities, system hardening, and risk management. Honeynets can be set up to model an entire operational network in order to research security risks and vulnerabilities of the network architecture.

(U) The Honeynet Project is an ongoing research effort that is conducted on a volunteer basis by a non-profit research organization of security professionals. The organization is dedicated to learning the tools, tactics, and motives of the blackhat community and sharing the lessons learned to benefit both its members and the security community. Founded in October 1999, the Honeynet Project is now in its fourth phase, which is to create a centralized system that can collect and correlate data from distributed honeynets.

2.6.3.2.2 (U) Usage Considerations

2.6.3.2.2.1 (U) Implementation Issues

(U) The two legal issues that need to be addressed when deploying deception technologies are entrapment and privacy. Although some attackers would like to argue that their activity was induced or persuaded, this is not the case. Attackers target honeypots/honeynets on their own initiative. Therefore, entrapment is most likely not an issue.

(U) When deploying deception technologies in the U.S., three legal issues must be considered:

- (U) Ensure compliance with laws restricting your right to monitor activities of users on your system
- (U) Recognize and address the risk that the honeypot may be misused by attackers to commit crimes, or store and distribute contraband
- (U) Consider the possibility that the honeypot can be used to attack other systems and result in potential liability for damages

(U) At the federal level, the two main statutes concerning communications privacy are the Electronic Communication Privacy Act (18 USC 2701-11) and the federal Wiretap Statute (Title III, 18 USC 2510-22). Outside of the U.S., the applicable laws of jurisdiction may be different and should be investigated further.

(U//FOUO) Honeypot and honeynet implementations can be complex and will vary depending upon the specific goals and objectives. Honeypots should be placed behind the firewall protecting the operational systems in order to mitigate risk. By doing so, the firewall will be able to log all traffic going through it and can provide some initial alerting capability. Review of the firewalls logs, assuming the firewall is not compromised, will assist in determining how the attack was initiated. Any packets sent to the honeypot are most likely probes from an attacker as no one should be communicating with it. Any traffic from a honeypot is indication that the device has been compromised. This is where it is critical to have the honeypot behind a firewall—to strictly control traffic to and from the honeypot.

(U//FOUO) The system logs of the honeypot must be protected. An attacker will attempt to delete or modify system logs to cover their trail. In addition to normal system logs for the benefit of the attacker, provision must be made to export the real system logs (the ones tracking the attacker's moves) to a protected system for analysis. This has to be done in such a manner that a sniffer used by the attacker would not detect the log files were being sent. Different protocols and mechanisms can be used to achieve this.

(U//FOUO) A sniffer, running on the firewall, can be used to capture keystrokes and screen shots so that there is documentation of everything the attacker enters and sees. To prevent the hacker from using encryption to hide activities, all services such as Secure Shell (SSH) should be disabled.

2.6.3.2.2.2 (U) Advantages

(U) The simple concept of honeypots and honeynets give way to some powerful strengths and advantages:

- (U) Intrusion detection capability: Honeypots provide detection of new types of attacks (also known as "zero-day" attacks) that were undetected by other security mechanisms
- (U) No false positives: Honeypots, by nature, do not conduct authorized activity. Therefore, any activity captured by a honeypot is considered suspect
- (U) Small data sets of high value: Honeypots collect only small amounts of valuable information (i.e., what the attacker is doing and how the operational systems can be better protected), thus reducing the noise that needs to be analyzed
- (U) Divert and control: Attackers probing a network will encounter honeypots that divert activity away from operational systems during some percentage of the time. The time an attacker spends investigating a honeypot will delay an attack on a real system
- (U) Encryption or IPv6: Unlike most other security technologies, honeypots are unaffected by encrypted or IPv6 environments

(U) Low-interaction honeypots have the advantage of simplicity. These honeypots are typically easier to deploy and maintain with minimal risk. A plug-and-play approach that involves installing software, and selecting the operating systems and services to be emulated and monitored makes deployment easy for most organizations. In addition, by containing the attacker's activity by emulated services, the risk is mitigated by never allowing access to an operating system to attack or do harm. Low-interaction systems work well because any access is anomalous.

(U) High-interaction honeypots give the advantage of providing attackers with actual—not emulated—operating systems and services to interact with. This allows extensive amounts of information to be captured and as a result, a greater opportunity to learn the full extent of the attacker's behavior. Another advantage of high-interaction honeypots is that no assumptions on how an attacker will behave are made. Since the environment is open and all activity is captured, these honeypots are able to learn behavior beyond what is expected.

2.6.3.2.2.3 (U) Risks/Threats/Attacks

(U//FOUO) There are both security and liability risks involved with deploying honeypots. These devices will be compromised and could be used as launching points for other attacks. Given the fact that there are ways to fingerprint many honeypot implementations, it is safe to assume that an attacker will indeed determine that the device is a honeypot. Therefore, one must consider the threat that an attacker might retaliate in some way after being duped.

(U//FOUO) All honeypots can and will be detected by an attacker who lingers long enough. Some honeypots provide signatures that can be easily fingerprinted warning attackers to move on. The firewalls providing some of the analysis data and protecting the operational systems can and will be compromised by determined attackers. The honeypots themselves will eventually be compromised by attackers who gain root access to the systems. The primary risk is that an attacker takes control of the honeypot, or honeynet, and uses it against the remaining operational systems or uses it as a launch point to other systems.

(U//FOUO) Finally, there is a risk of overdependence on honeypots/honeynets. Although a honeypot/honeynet may be able to catch an attacker who is blindly groping a system, the same success will not be shared by a more sophisticated attacker with a focused mission. Therefore, implementing a honeypot/honeynet system may provide a false sense of security.

2.6.3.2.3 (U) Maturity

(U//FOUO) Honeypot technology has been around for many years and both commercial and Government-developed solutions are available. The current thrust in honeypot technology is to develop scalable solutions that more fully recreate a full operating system appearance to the attacker. In this regard, virtual machines have become highly useful to honeypot developers. Overall, maturity of honeypot technology is rated as Emerging (TRL 4-6), while honeynet technology is rated as Early (TRL 1-3).

2.6.3.2.4 (U) Standards

(U) There are no standards for honeypots and honeynets per se. However, there are standards that apply to data capture (what data should be captured at each honeynet and in what format) and data collection (what data should be sent to a central collection site and in what format).

(U) Data Capture Standards

- (U) All network activity (packets and full packet payload) must be captured in tcpdump binary format (OpenBSB libpcap standards) and rotated/compressed (gzip) on a daily basis
- (U) Firewall logs must be converted to ASCII format to allow uploading into a centralized database
- (U) An attacker's activity must be captured on the system itself. In the past, sniffing connections to capture keystrokes off the wire would suffice. However, attackers today are likely to adopt some form of encryption to communicate. The Honeynet Project has developed Sebek2, a kernel module that is capable of logging an attacker's keystrokes and capturing files uploaded via secure copy (scp)

(U) Data Collection Standards

- (U) Tcpdump binary logs – each honeynet can forward daily tcpdump binary log captures
- (U) Firewall logs – every inbound and outbound connection logged by the firewall can be sent in ASCII text format on a daily basis

2.6.3.2.5 (U) Cost/Limitations

(U) Deception-based technologies are not necessarily expensive to deploy. The cost is dependent upon the size of the operational system in which they are being placed and the maintenance and support cost to operate and manage them. First and foremost, it is important to consider the nature and cost of containment and control. Measures should be taken to mitigate the risk of having a honeypot system deployed in a network. If a product does not support any native containment and control, the cost and complexity of implementation should be seriously examined.

(U) Analysis of the data is another cost that must be factored. Some products provide integrated analysis, reporting, and alerting. However, other products require involvement by an administrator, which could have a significant impact on the cost of using such a system. Ongoing administrative costs include maintenance of content and restoration of the honeypot. Periodic updates to the content will be essential to maintain the appearance of a valid and live system. Also important is the need to periodically restore the system to a clean and controlled state. Once again, automated capabilities for restoration can greatly reduce administrative costs.

(U) Honeypots, like any other technology, have limitations. Honeypots have a limited view in that only activity with direct interaction can be tracked and captured. Therefore, attacks made against other systems will not be captured. Also, chances are that an attacker will eventually learn that a device is a honeypot and either leave after cleaning up as much as possible, or worse, take punitive action against the operational systems. Even a successful honeypot will provide valuable data on the steps taken by an attacker, which has to be delivered to another system without the attacker's knowledge and then undergo extensive analysis before it can prove to be useful.

2.6.3.2.6 (U) Dependencies

(U) As an information gather tool, a honeynet can employ Methodology Fingerprinting to determine the patterns of behavior of a particular attack or attacker, as well as be used to discover the unknown.

(U) A honeynet can perform these tasks by controlling, capturing, and analyzing data. Data control involves such activities as restricting inbound and outbound traffic from a compromised honeynet. Such tools as a Honeywall, firewalls such as OpenBSD firewall and Snort_inline (a modified version of Snort that is used in the Honeynet Project to drop or modify packets) would handle data control.

(U) Capturing data allows the honeynet analysts to observe intruders, even in encrypted environments and without being noticed by the intruder. The honeynet analysts can also monitor all attacker activity. Data can be captured via keylogging, firewall logs, packet sniffer logs and Honeyd logs to name a few. Snort, Sebek, and Termlog are a few tools that may be used to capture data within a honeynet. The data can then be exported to a server for analysis.

(U) Data analysis includes traffic analysis (IP addresses and ports, traffic frequency and volume), fingerprinting (flags and options indicate platform personality), content analysis, granularity, confidentiality issues, encryption, and digest analysis. Examples of tools used to analyze the data include HoneyInspector, which enables real-time analysis, PrivMsg, which extracts IRC conversations from tcmdump binary log files (eliminates noise), and Sleuthkit, a forensic toolset for analyzing hacked systems.

(U) Most of the technologies (workstations, servers, firewalls, etc.) used to create Honeynets are not new. However, many of the tools used to control, capture, and analyze the data are new. Tools such as Sebek have become available within the last two years. In the future, developers are planning to include capabilities to automatically filter large volumes of data, correlate IDS data with other network data, and provide a unified view of the event or attack.

2.6.3.2.7 (U) Alternatives

(U) Other capabilities exist that could track the activities of attackers, but none in so controlled an environment.

2.6.3.2.8 (U) Complementary Techniques

(U) Honeypots complement network-based and host-based Intrusion Detection Systems (IDSs). Although closely related, honeypots do not require the capability to discriminate between operational traffic and attacker traffic nor share the likelihood of many false positives.

2.6.3.2.9 (U) References

(U) "The Evolution of Deception Technologies as a Means for Network Defense," Recourse Technologies, http://www.sans.org/rr/papers/30/recourse.php.

(U) "Honeypot Definitions, Requirements, Standards," version 1.5.3, 22 October 2003, http://www.honeynet.org/alliance/requirements.html.

(U) "Honeypots - Definitions and Value of Honeypots," by Lance Spitzner, http://www.tracking-hackers.com/papers/honeypots.html, 29 May 2003.

(U) "Honeypots - Definitions and Value of Honeypots," by Lance Spitzner, http://www.secinf.net/honeypots/Honeypots_Definitions_and_Value_of_Honeypots.html, 10 December 2002.

(U) "Honeypots, the Hottest Thing in Intrusion Detection," by John Harrison, http://channelzone.ziffdavis.com/article2/0%2C1759%2C1516562%2C00.asp, 4 November 2003.

(U) "The Honeynet Project," http://www.honeynet.org/misc/project.html.

(U) "Honeynet Project: What a Honeynet Is," http://www.awprofessional.com/articles/printerfriendly.asp?p=23948.

(U) "Know Your Enemy: GenII Honeynets," http://www.securesynergy.com/library/articles/051-2003.php.

(U) "Know Your Enemy: Honeynets," Honeynet Project, http://www.honeynet.org/papers/honeynet/index.html, 12 November 2003.

(U) "Know Your Enemy," Chapter 8: Legal Issues, by Richard Salgado, http://www.honeynet.org/book/Chp8.pdf, 29 April 2004.

2.6.3.3 (U) Situational Awareness

2.6.3.3.1 (U) Technical Detail

2.6.3.3.1.1 (U) UDOP

(U//FOUO) Network situational awareness capabilities include monitoring tools (network health, bandwidth utilization, and key servers and processes) on-hand as percentage of those required for fixed and deployed forces. The CND UDOP provides situational awareness of CND activities, operations, and their impact, collaboration, and decision support to all levels of the GIG. The CND UDOP is the integration of a comprehensive data presentation interface and data storage coupled with intelligent data acquisition. The resulting solution is robust and flexible and provides situational awareness information across the DoD to support the Warfighter.

(U//FOUO) These basic requirements define what will be included in the CND UDOP. The CND UDOP is defined as that portion of the IA and Network Operations (NETOPS) operational picture that provides local, intermediate, and DoD-wide situational awareness of CND activities, operations, and their impact, collaboration, and decision support. The emerging CND UDOP leverages common data, views, and mechanisms for data sharing and displays all information necessary for the defense of DoD networks.

2.6.3.3.1.2 (U) NETOPS

(U//FOUO) The CND UDOP is expected to receive the majority of its data from sources that will also feed the larger NETOPS picture.

(U//FOUO) Information used to support the UDOP consists of both raw data inputs and processed and correlated alert information. Flow data is currently used for a number of analytical techniques—namely scan and application detection. Many analysis methods are available, and many others are under development.

(U//FOUO) Core routers form the backbone of the existing monitor network. Attack detection and prevention systems installed at the core routers have the potential to detect and block attacks before they reach the enclaves. Sensors at these locations provide the analyst with a high-level view of attacks launched against large numbers of hosts located at different physical locations (provided that the data from the sensors is aggregated at some point). Also, a small number of sensors are required to detect and block attacks against a large number of hosts.

(U//FOUO) Analysis is conducted on both raw and processed data whether acquired from the existing sensor grid or from other sources. The analysis uses both automated and manual means to correlate sensor grid data, alarms, and event detections. An alarm management interface provides operators the ability to acknowledge alarms and perform COAs on those alarms. When launched from the main dashboard level, the interface can show all of the alarms for the operator’s sphere of responsibility.

2.6.3.3.2 (U) Usage Considerations

2.6.3.3.2.1 (U) Implementation Issues

(U//FOUO) Usage considerations are complex and varied. The following list identifies significant requirements the system must deliver to the user.

- (U//FOUO) The system must accomplish information sharing and information/data transmission in an appropriately controlled and secure environment, ensuring the appropriate security classification level for each level of user
- (U//FOUO) In disseminating technical information to users, the system must provide the capability to evaluate, integrate, and synchronize proposed CND options with overall battle and security plans
- (U//FOUO) The system must disseminate information on defensive strategies to the CND community
- (U//FOUO) The system must provide information sharing and collaboration capabilities for near real-time tactical warning between the operations and intelligence communities
- (U//FOUO) The system must provide a capability for distributed collaboration to coordinate mitigation and response in execution of the CND mission
- (U//FOUO) The system must effectively enable controlled, releasable, and discloseable information sharing among authorized users within the DoD, other U.S. Government departments and agencies, law enforcement and other emergency response agencies, selected non-government and private sector entities, and organizations across a global architecture
- (U//FOUO) The system must be scalable and adaptable to dynamic user requirements and have the reserve capacity to support surge loading and multiple military operations

(U//FOUO) A sensor needs to be placed at every entrance and exit point to or from the network being protected. If the network in question has no gateways (LAN), an assessment must be made as to what the best collection points on the network are.

(U//FOUO) The use of multiple and diverse sensor products compounds the analytical task of the network analysts. Each sensor has its own unique method for analyzing network packets and network sessions and for determining what constitutes an alert. To reduce the number of alerts sent to the analysts from different sensors, an automated approach to data correlation and summarization is needed.

(U//FOUO) Data correlation needs to occur across commercial and government products, bridging the gap between network sensors, host-based information, and audit logs. Flow data is centralized and used to detect patterns. Mechanisms for centralizing data (dedicated circuits) must be in place to transport this data. The automated analysis of attacks should include indications of severity levels, damage assessment, and recommended COAs.

2.6.3.3.2.2 (U) Advantages

(U//FOUO) Effective CND requires an operational view of the networked environment to provide situational awareness of potential threats, attacks, network status, and other critical information to support a mission commanders' decision-making and prevent, stop, or reverse degradation of network resources due to unauthorized activities. The criticality of enhancing CND situational awareness is due to the increasingly information-centric operations conducted by DoD and its allies. Specifically:

- (U//FOUO) Commanders and their forces are dependent upon accurate, complete, reliable, timely, and secure information to conduct their missions
- (U//FOUO) Commanders and their forces are dependent on the GIG and other assets, and need to know when situations exist that can affect the information systems and networks supporting their critical Warfighting processes
- (U//FOUO) DoD must protect, monitor, detect, analyze, and respond to unauthorized activity within DoD information systems and global networks to ensure continuity of operations throughout the spectrum of conflict (i.e., CND)
- (U//FOUO) Commanders need the capability to quickly comprehend the status and reliability of their information and information systems to successfully engage in network centric operations
- (U//FOUO) Network operators need the capability to develop user defined operational pictures (UDOP) for tailored or filtered views to meet the specific needs of Commanders and deployed Warfighters
- (U//FOUO) Network operators require insight into the networked environment that will permit real-time decisions supporting security, continued availability, and restoration of DoD networks
- (U//FOUO) Network operators need the capability to quickly share information concerning the status of allied/coalition nations' Command, Control, Communications, Computers (C4) systems

(U//FOUO) Another advantage of the envisioned centralized structure is the use of flow analysis. Flow analysis requires much less data than content analysis, which eases computing, data transfer, and data storage requirements, resulting in significant performance benefits on a global network such as the GIG.

2.6.3.3.2.3 (U) Risks/Threats/Attacks

(U//FOUO) The CND UDOP is expected to get the majority of its data from sources deployed in the ESG. However, additional data sources (Joint CERT Database, Indications & Warnings, etc.) will need to be used to complete the CND UDOP picture. The correlation of information from many distributed sources represents both a risk and a challenge for DoD.

(U//FOUO) The collaboration engine and tools are the hooks into the DoD collaboration software that among other capabilities allows users of the UDOP to collaborate with other users. The collaboration interface will include the capability for users of the UDOP to share and discuss incidents, reports, and alarms. Any failure of this collaboration capability could significantly impact mission success.

2.6.3.3.3 (U) Maturity

(U//FOUO) Automated Indications and Warning (I&W) is a proactive process that involves collecting, assembling, and analyzing large amounts of intelligence data from a variety of sources. Current collection, correlation, and visualization capabilities exist that support a NETOPS. The CENTAUR flow data analysis system has been operational since 2000.

(U//FOUO) The rate of increase in network bandwidth is currently greater than the rate of increase in processing speeds and the rate of increase of memory sizes and speeds. As a result, automated I&W components built in software (i.e., IDSs, Traffic Normalizers [TNs], and Intrusion Prevention Systems [IPSs]) face significant difficulty being able to handle traffic at full line rate. Unfortunately, creating custom hardware such as Application-Specific Integrated Circuits (ASICs) requires a significant investment in manpower and in planning. In response to this need, various vendors have created programmable embedded systems that can process packets at full line rate in Gigabit, or higher, networks. Such technologies are broadly referred to as Network Processors (NPs).

(U//FOUO) Overall, the maturity levels of both UDOP and NETOPS technologies are rated Emerging (TRL 4-6).

2.6.3.3.4 (U) Standards

(U//FOUO) Other then DoD Information Technology Security Certification and Accreditation Process (DITSCAP)-related certification and accreditation requirements, there are no standards directly applicable to this technology area. However, a requirements document, Computer Network Defense User Defined Operational Picture (CND UDOP) Requirements List, 23 March 2004, has been released. This requirements list is expected to influence emerging standards by providing recommendations on a vendor-neutral, sensor information exchange format and interface standard.

2.6.3.3.5 (U) Cost/Limitations

(U//FOUO) Analysis data centers are affordable. Cluster technology, which combines independent computers into a unified system (or cluster) through software and networking, makes this analysis extremely scalable. Clusters are typically used for high availability to provide greater reliability or high performance computing to provide greater computational power than a single computer can provide. Beowulf clusters are an example.

(U) Limitations to integrating a complete UDOP include the implementation of an enhanced sensor grid across the DoD enterprise, developing technologies scalable to the GIG, and creating detection tools that work with the IPv6 protocol. Sensor development is ongoing and will be deployed at some level in the near future. However, to achieve the full vision of the UDOP, more robust sensors will be necessary. Implementation of the sensor grid and continued research into the gap areas will further extend the current UDOP capabilities to meet UDOP needs.

2.6.3.3.6 (U) Dependencies

(U//FOUO) Visualization and correlation engine capabilities are dependent on both the ability to collect data from many sources at high data rates, and the ability to analyze this data in near real time. This technology requires a source of flow data, bandwidth to centralize data, and sufficient disk storage to store and process data. The ESG of 2008 is envisioned as a grid of sensors, each fully capable of collecting sufficient data in near real time to meet the needs of the UDOP. Fully implementing the sensor grid and maintaining sufficient centralized storage capacity are critical collection capabilities.

2.6.3.3.7 (U) Complementary Techniques

(U//FOUO) A complementary technique exists on the DoD enterprise at this time. CENTAUR is a metadata collection, storage, and analysis system that accepts Netflow data as its primary input. It enables analysis of traffic flow data produced by routers to determine the presence of malicious activity. The information is used to correlate/collaborate both reported incidents, as well as to detect anomalous activity including blatant and stealthy activities. Operational incidents are reported, and correlation of various network data is performed, reported, and distributed accordingly.

2.6.3.3.8 (U) References

(U) “Assessment of IA/CND Focus Areas,” Mitre Corporation, June 2004.

(U) “Beowulf Project Overview,” http://www.beowulf.org/overview/index.html.

(U) “Computer Network Defense User Defined Operational Picture (CND UDOP) Requirements List,” DISA, 23 March 2004.

2.6.3.4 (U) Network Mapping

2.6.3.4.1 (U) Technical Detail

(U//FOUO) Vulnerability scanning tools can discover and store topology and status information about transport-layer optical devices to data routers, switches, and IP addresses. They also have the capability to conduct a basic mapping of applications to their underlying systems and servers. These tools provide a graphical view of the environment and provide indicators of the presence of new devices that have appeared since the last scan.

(U//FOUO) Vulnerability management tools find, evaluate, and optionally, eliminate vulnerabilities on systems before attackers take advantage of them. Efficient and comprehensive, near-real-time discovery tools are needed for accurate analysis of DoD's physical and virtual networks, as well as to identify applications running on the network and manifestations of cyber situational awareness. These tools are also needed to ensure that users adhere to security policies and to deter users from introducing vulnerabilities. This desired capability must also be scalable to very large networks.

(U//FOUO) In some applications, mappers are combined with vulnerability databases and other correlation tools to identify potential weaknesses or routes of attack for various components. In such cases, these posture discovery tools enhance:

- (U) Security Monitoring/Management
- (U) Network Security
- (U) Problem Management

2.6.3.4.2 (U) Usage Considerations

(U//FOUO) Usage considerations relate to the legal concerns associated with either passive listening or active vulnerability identification. Passive discovery tools have virtually no impact on normal operations as they represent one-way listening devices on the network. Active discovery tools impact bandwidth availability and can cause intrusion detection alarm conditions.

2.6.3.4.2.1 (U) Implementation Issues

(U//FOUO) In order to work most effectively, vulnerability management tools should have unimpeded access to the systems to be tested. Therefore, the vulnerability management tools must be inside of any site firewalls. In cases where multiple subnets protected by separate firewalls exist within an enclave, multiple vulnerability management tools will be needed. This increases the number of tools required—increasing cost and management difficulties.

2.6.3.4.2.2 (U) Risks/Threats/Attacks

(U//FOUO) The very feature of the GIG that makes it most beneficial, the ubiquitous access to the system resources enterprise-wide, is also what makes the GIG an attractive target for adversaries. There are over 90 countries with affirmed nation state computer network exploitation efforts at the tactical and strategic levels, most collecting critical information against the United States and her allies. Nearly 20 countries have confirmed, dedicated computer network attack programs. Their mere existence suggests success and satisfaction with the returns on their investments.

(U//FOUO) In the GIG vision, all systems will be interoperable and information of all types reachable from anywhere. As a result there will be many insiders with potential access to information that was not available to them before. In addition, the connection of temporary coalition partners to the GIG will widen system access beyond our immediate control. Without accurate vulnerability detection and control, an adversary can use this increased capability against us and/or deny us the use of these capabilities at the point when we have become most reliant upon them.

2.6.3.4.3 (U) Maturity

(U//FOUO) Current network mapping and vulnerability discovery approaches are used for configuration management, vulnerability reduction, and for resource identification in large-scale enterprise networks. While both active and passive mapping solutions are currently installed, the usual means of accurate network discovery is to actively perform port mapping and open port investigations on network components to determine the following:

- (U) The current host operating system
- (U) The primary use of the network component
- (U) Services and applications running on the system
- (U) The physical connections and topology of the network
- (U) If current platforms have unpatched vulnerabilities or are running unsecured services

(U//FOUO) Together, the maturity of the various technologies of the Network Mapping technology area is rated as Emerging (TRL 4-6).

2.6.3.4.4 (U) Standards

(U//FOUO) Standards for mapping and vulnerability discovery are not applicable. In the GIG environment, near continuous monitoring will be essential because the environment will be so dynamic (ad hoc creation of expedient COIs and the associated vulnerabilities, etc.). Alert on Change could more appropriately be called Alert and Change since automated decision-making and response, based on GIG-wide situational awareness, will be vital to provide Active Network Defense for the GIG. This will require the use of agents and a GIG-wide hierarchy of agent functional stacks for correlating and fusing the data from homogeneous and heterogeneous sensor agents spread throughout the GIG. In this regard, standards should be developed to reflect automated agent-based change mechanisms.

(U) Common Vulnerabilities and Exposures (CVE) is a list or dictionary that provides common names for publicly known vulnerabilities and information security exposures. CVE standardized the names for all publicly known vulnerabilities and security exposures.

(U) Open Vulnerability Assessment Language (OVAL) is the common language for security experts to discuss and agree upon technical details about how to check for the presence of vulnerabilities on computer systems. OVAL queries are based primarily on the known vulnerabilities identified in CVE.

2.6.3.4.5 (U) Cost/Limitations

(U//FOUO) Mapping and vulnerability tools themselves represent minor costs. However, the implementation of an agent-based discovery technology, including the capability for automated vulnerability repair, represents both costs and limitations based on acceptance of allowable command functions. Further, although agent-based solutions require greater deployment labor, they ultimately reduce operating cost by enabling near-continuous monitoring and potential Alert and Change.

2.6.3.4.6 (U) Dependencies

(U//FOUO) To be effective in a large and ever changing environment, the preferred discovery approach needs to be both fast and focused. It should interpret the initial conditions and transpose these conditions through a scalable interface to the system administrator. Discovery tools should also have the inherent capability to interface with other tools and agent-based architectures at local host and hierarchal levels so more advanced discoveries or controls can take place. The ultimate goal would be to identify and then correct the identified deficiencies. Such response tools are dependent on both accurate discovery and automated decision making.

(U//FOUO) The GIG's Net-Centric Operations Warfare (NCOW) development problem related to vulnerability discovery is best viewed not just as communications, networking, information assurance, and knowledge dissemination problem. Rather, due to size and complexity, it should be viewed primarily as an artificial intelligence (AI) problem. Here, artificial intelligence is defined as:

> (U) A collection of algorithms and their attendant infrastructure organized to automate a decision-making process.

(U//FOUO) The GIG will need a ubiquitous AI architecture to address the discovery problem. The key expected AI building block, an Agent Functional Stack (AFS), is a collection of specialized intelligent agents organized into layers, thus providing services to each other. These should be used for managing IA services at the enclave initially and in the future at a COI.

2.6.3.4.7 (U) Alternatives

(U//FOUO) Many network mapping programs are designed to automatically discover a local network. They use SNMP or pings to identify network devices and work out how they are physically connected together. The network is then presented as a topology diagram with simple integrated monitoring. Changes in the network are reflected in the diagram that continuously updates and are usually customizable to provide various views of the network map. Some of these tools identify the characteristics of the links as well as the various devices, including their operating system, primary use, and some even what ports they have open. While current graphical network monitoring can be a useful management tool for system administrators, active monitors can become bandwidth intensive when used in enterprise-level applications.

(U//FOUO) A probe usually implements a portion of the device's protocol; if the device does not answer properly, the mapper will indicate the device as down, or in the alarm or warning state. Devices on the network are usually shown as various figures on a map. In general, each figure represents a piece of network equipment (such as a router, switch, or hub), a workstation, a database, or a server.

2.6.3.4.8 (U) Complementary Techniques

2.6.3.4.9 (U) References

(U) “Introduction to Vulnerability Scanning,” by Tony Bradley, Internet/Networking Security, http://netsecurity.about.com/cs/hackertools/a/aa030404.htm.

(U) http://www.cve.mitre.org/

(U) http://www.us-cert.gov/oval/about.html

2.6.3.5 (U) Intrusion Detection Systems

2.6.3.5.1 (U) Technical Detail

(U) Due to the ubiquitous nature of the Internet and local networks today, organizations have seen an increase in the number of systems being implemented that can monitor against the growing number of intrusion events and security breaches. While IDSs are primarily concerned with the detection of hostile actions, they can, in some cases, even initiate a series of actions in response to an intrusion or attack.

(U) The two primary types of IDSs are:

- (U) Host-Based IDS – derives its source of information from a single host or system
- (U) Network-Based IDS – derives its source of information from a whole segment of a local network

2.6.3.5.1.1 (U) Host-Based IDS (HIDS)

(U) A HIDS resides on a single computer and provides protection for that specific computer system. This allows HIDS to analyze activities with great reliability and very fine granularity. HIDS are essential in monitoring systems that reside on high-speed networks and in networks in which encryption is used. As HIDS are used in more critical locations, their ability to both detect and withstand attacks must also increase. By implementing a HIDS within the kernel layer, detection can be placed closer to the root operation that compromises a system. This improves the ability of a HIDS to detect both known and unknown attacks. Implementation within the kernel can also help maintain the robustness of the HIDS itself by having it run within protected space where it cannot be easily modified or subverted.

(U) The current state of technology for kernel-layer HIDS shows an increasing emergence of COTS products. Most of these products are agent-based solutions. They intercept system calls between applications and the kernel, but do not run within the context of the kernel. This would make them more vulnerable to mimicry attacks and attacks against their availability.

2.6.3.5.1.2 (U) Network-Based IDS (NIDS)

(U) The majority of commercial intrusion detection systems are network-based. These IDSs detect attacks by capturing and analyzing network packets. By listening on a network segment or switch, one NIDS can monitor the network traffic affecting multiple hosts that are connected to the network segment, thereby protecting those hosts. The need to work online with encrypted networks destined to a single host has seen the introduction of what some consider a separate class of intrusion detection systems, known as Network Node IDS (NNIDS). NNIDS are a blend of HIDS and NIDS, with agents deployed on every host within the network being protected (typical NIDS uses network agents to monitor whole LAN segments). Most of the large intrusion detection systems that are commercially offered today merge the strengths of HIDS and NIDS into a unique concept.

(U) There are generally two different analysis approaches of IDSs: misuse detection and anomaly detection.

- (U) Misuse Detection - Misuse detection techniques attempt to model attacks on a system as specific patterns, and then systematically scan the system for occurrences of these patterns. This process involves a specific encoding of previous behaviors and actions that were deemed intrusive or malicious. Misuse detectors analyze system activity, looking for events or sets of events that match a predefined pattern of events that describe a known attack. As the patterns corresponding to known attacks are called signatures, misuse detection is sometimes called signature-based detection. The most common form of misuse detection used in commercial products specifies each pattern of events corresponding to an attack as a separate signature. However, there are more sophisticated approaches called state-based analysis techniques that can leverage a single signature to detect groups of attacks. After revamping the commercial IDSs with signatures which reflect generic attack classes, we seem to be in a very good position to detect incoming attacks through content examination.

- (U) Anomaly Detection - Anomaly detection approaches attempt to detect intrusions by noting significant departures from normal behavior. Anomaly detectors identify abnormal unusual behavior (anomalies) on a host or network. They function on the assumption that attacks are different from normal (legitimate) activity and can therefore be detected by systems that identify these differences. Anomaly detection, while initially attractive, has yet to show any promise due to the large number of false alarms that are created. Although some commercial IDSs include limited forms of anomaly detection, few, if any, rely solely on this technology. The anomaly detection that exists in commercial systems usually revolves around detecting network or port scanning. However, anomaly detection remains an active intrusion detection research area and may play a greater part in future IDSs.

2.6.3.5.2 (U) Usage Considerations

2.6.3.5.2.1 (U) Implementation Issues

(U) Current systems require full content examination. However, metadata-based detectors have shown promise in handling scans and large-scale worm activities. This means that detection is still very much content-oriented and that future detectors must continue to be able to handle full content examination.

(U) Physical limitations dominate. Sufficient cooling, power, and rack space requirements are the driving factors. Load balancing is also a must as detectors have fixed bandwidth limitations.

(U) There are serious concerns about deployment of NIDS and HIDS in the GIG. The current architectures used by DISA and the services NIDS may not be compatible with the Black Core concept and may not scale well. Further, there is little protection at the enclave level today. HIDS are rarely used, if at all, and planning and deployment in the GIG requires significant architectural work. Subsequently, the integration of many NIDS and HIDS into the tiers envisioned for the GIG will require significant architectural work, standards development, and technology development.

2.6.3.5.2.2 (U) Advantages

(U) Host-based IDSs, with their ability to monitor events local to a host, have the advantage of being able to detect attacks that cannot be seen by a network-based IDS. Also, host-based IDSs can often operate in an environment in which network traffic is encrypted by gathering information before data is encrypted or after the data is decrypted at the destination host.

(U) The advantages of misuse/signature detection methods are:

- (U) Lower false alarm rates
- (U) Simple algorithms
- (U) Easy creation of attack signature databases
- (U) Easy implementation
- (U) Typically minimal system resource usage

(U) The advantages of anomaly detection methods are:

- (U) Possibility of detection of novel attacks
- (U) Anomalies are recognized without specific knowledge of details
- (U) Ability to detect abuse of user privileges
- (U) Ability to produce information that can in turn be used to define signatures for misuse detectors

2.6.3.5.2.3 (U) Risks/Threats/Attacks

(U) A risk exists in the fact that current commercial IDSs do not detect novel attacks, nor do they catch most novel variations of attacks. This is a significant technology gap in the IDS technology area and calls for more research and development.

(U) Furthermore, an important distinction needs to be made between the terms: detection and recognition. For signature-based systems, there is very little difference between the two—detection means that an attack is recognized, at least for those systems with very low false-positives. However, the same is not true for anomaly-based systems. Here detection data that has been recorded may not result in a report or recognition, but still be analyzed more deeply at a later time. Misuse detectors do not report or record near misses and so the only time the detection data is available is when an attack is recognized.

2.6.3.5.3 (U) Maturity

(U) While much IDS research is underway, commercial IDSs are still in their formative years. The negative publicity of some commercial IDSs can be attributed to the following:

- (U) Large number of false alarms
- (U) Awkward control and reporting interfaces

- (U) Overwhelming number of attack reports
- (U) Lack of scalability
- (U) Lack of integration with enterprise network management

(U) However, there is a strong commercial demand for IDSs that will increase the likelihood of these problems being addressed in the near future.

(U//FOUO) The various sub-technologies of the Intrusion Detection System technology area can be generally assigned Technology Readiness Level group of Emerging (TRL 4-6).

2.6.3.5.4 (U) Standards

(U) Standardization is problematic as we are still dependent on vendor hardware that changes from time to time. Still, sensor outputs are standardizing with almost everyone supporting Packet Capture (PCAP). The PCAP library provides a high level interface to packet capture systems and allows access to all packets on the network.

(U) There are no mature IDS standards at this point in time. The Internet Engineering Task Force (IETF) has one working group, the Intrusion Detection Working Group (IDWG), which is tasked with defining "formats and exchange procedures for sharing information of interest to intrusion detection and response system, and to management systems which may need to interact with them." The standards are listed in Table 2.6-1

Table 2.6-1: (U) Standards for Intrusion Detection Systems

This table is (U)	
IETF Intrusion Detection Working Group (drafts)	
Name	**Description**
Intrusion Detection Message Exchange Requirements (October 22, 2002)	This Internet-Draft describes the high-level requirements for sharing IDS information. http://www.ietf.org/internet-drafts/draft-ietf-idwg-requirements-10.txt
The Intrusion Detection Message Exchange Format IDMEF (January 8, 2004)	Describes a data model to represent information exported by intrusion detection systems, and explains the rationale for using this model. http://www.ietf.org/internet-drafts/draft-ietf-idwg-idmef-xml-12.txt
The Intrusion Detection Exchange Protocol (IDXP) (October 22, 2002)	Describes the Intrusion Detection Exchange Protocol (IDXP), an application-level protocol for exchanging data between intrusion detection entities. http://www.ietf.org/internet-drafts/draft-ietf-idwg-beep-idxp-07.txt
IETF Incident Handling Working Group (working drafts)	
Distributed Denial of Service Incident Handling: Real-Time Inter-Network Defense (February 28, 2004)	This proposal integrates current incident detection and tracing practices for network traffic, which could be extended for security incident handling. Policy guidelines for handling incidents are recommended and can be agreed upon by a consortium using the defined protocol and extended to each NP's clients. http://www.ietf.org/internet-drafts/draft-moriarty-ddos-rid-06.txt
The Incident Object Description Exchange	Provides implementation guidelines for Computer Security Incident Response Teams (CSIRT) adopting the Incident Object Description Exchange Format

This table is (U)	
Format (IODEF) Implementation Guide (March 9, 2004)	(IODEF). http://www.ietf.org/internet-drafts/draft-ietf-inch-implement-00.txt
This Table is (U)	

(U) Preliminary implementation work is probably possible, though implementations would have to change as the standard is finalized. The design involves sending XML-based alerts over an HTTP-like communications format. A lot of attention has been paid to the needs of IDS analysis and to making the protocol work through firewalls in a straightforward way.

(U) There is also an effort by the ISO's T4 committee to develop an Intrusion Detection Framework. The status of that effort is presently unknown, and attempts to gather further information have been unsuccessful.

(U) The following Protection Profiles have been evaluated and certified with NIAP (http://niap.nist.gov/cc-scheme/pp/index.html):

- (U) Intrusion Detection System (System) Protection Profile, Version 1.4
- (U) Intrusion Detection System (Analyzer) Protection Profile, Version 1.1
- (U) Intrusion Detection System (Sensor) Protection Profile, Version 1.1
- (U) Intrusion Detection System (Scanner) Protection Profile, Version 1.1

2.6.3.5.5 (U) Cost/Limitations

(U) The acquisition of IDS software is not the actual cost of ownership. Additional costs include acquisition of a system to run the software, specialized assistance in installing and configuring the system, personnel training, and maintenance costs. Personnel to manage the system are the largest cost.

(U) Most host-based systems implement common architectures in which a host system works as a host agent reporting to a central console. The associated costs of HIDS deployments can vary depending on vendor and software versions.

(U) Network-based systems can be deployed as stand-alone hosts with a possible management interface or distributed sensors and management console. The cost of commercially available sensors varies depending on vendor, bandwidth, and functional capabilities. Management consoles can be free or can cost several thousand dollars—also depending on the vendor. Additional costs include hardware or back-end databases.

(U) Intrusion detection technology is still evolving. Limitations of IDSs include the following:

- (U) Certain types of attacks are still possible which preclude detection at present
- (U) Bandwidth is a serious limitation on most hardware

2.6.3.5.6 (U) Dependencies

(U) We are still largely dependent on commercial vendors for hardware/basic software development. Other trends in computing that is believed will affect the form and function of IDS products include the move to appliance-based IDSs. It is also likely that certain IDS pattern-matching capabilities will move to hardware in order to handle increased bandwidth.

2.6.3.5.7 (U) Alternatives

(U) Government-developed IDSs may be better suited for generic attack class detection. Currently, systems rely on a commercially developed base which has been optimized to detect singular attacks.

(U) Current anomaly detection methods have proven inadequate, and therefore prompt new methods to be researched and tried. Additionally, very little research has been done in the area of parallel processing of content via Beowulf clusters even though this area shows much promise. Beowulf clusters are scalable performance clusters that are based on commodity hardware, on a private system network and with open source software (Linux) infrastructure. Each cluster consists of PCs or workstations that are dedicated to running high performance computing tasks, with improved performance being proportional to added machines.

(U) Traffic normalizers, commonly referred to as protocol scrubbers, are inline network devices that remove protocol ambiguities from network traffic. The primary objective of the traffic normalizer is to provide a security enhancement that aids in preventing insertion, DoS, and evasion attempts against IDSs, thereby eliminating a weakness of many IDSs.

2.6.3.5.8 (U) Complementary Techniques

(U) As stated earlier, metadata-based detection can take some of the load off content-based examination. However, in the end, some content must be made available to validate any attack.

(U) When used in conjunction with firewalls and other access control devices, IDSs can bolster an organization’s ability to detect, prevent, and respond to unauthorized access and intrusion attacks. Firewalls, if positioned within the enclave, can decrease the amount that an IDS is required to examine. In addition, any number of policy enforcement mechanisms (i.e., guards, OS/application wrappers, and anti-virus) can become complements to an IDS.

(U) Several other tools exist that are often labeled as intrusion detection products by vendors. These tools include vulnerability analysis/assessment systems, file integrity checkers, and attack isolation.

2.6.3.5.9 (U) References

(U) “Algorithm-Based Approaches to Intrusion Detection and Response,” by Alexis Cort, 16 March 2004.

(U) Beowulf Project Overview, http://www.beowulf.org/overview/index.html.

(U) “Defending Yourself: The Role of Intrusion Detection Systems,” by John McHugh, Alan Christie, and Julia Allen, IEEE Software – September/October 2000.

(U) "FAQ: Network Intrusion Detection Systems, by Robert Graham," accessed on 10 March 2004, http://www.robertgraham.com/pubs/network-intrusion-detection.html#2.1.

(U) "Intrusion Detection FAQ: What open standards exist for Intrusion Detection?," by Stuart Staniford-Chen, 8 April 2000, http://www.sans.org/resources/idfaq/id_standards.php.

(U) "Intrusion Detection: Implementation and Operational Issues, by Julia Allen," Alan Christie, and John McHugh, IEEE Crosstalk, January 2001.

(U) "Intrusion Detection Systems (IDS) Part 2 – Classification; Methods; Techniques," by P. Kazienko and P. Dorosz, 15 June 2004, http://www.windowsecurity.com/articles/IDS-Part2-Classification-methods-techniques.html.

(U) "Justifying the Expense of IDS, Part One: An Overview of ROIs for IDS," by David Kinn and Kevin Timm, Security Focus (Infocus), 18 July 2002.

(U) "NIST Special Publication on Intrusion Detection Systems," by Rebecca Bace and Peter Mell.

(U) "Why is a firewall alone not enough? What are IDSes and why are they worth having?," by Wojciech Dworakowski, 23 July 2004, http://www.windowsecurity.com/articles/Why_is_a_firewall_alone_not_enough_What_are_IDSes_and_why_are_they_worth_having.html.

2.6.3.6 (U) Intrusion Prevention Systems (IPSs)

2.6.3.6.1 (U) Technical Detail

(U) A new category of CND technologies has recently emerged in the COTS environment. Called Intrusion Prevention Systems (IPSs), these technologies represent the merger of protect capabilities with intrusion detection capabilities. As technology advanced, it became clear that if a device knowingly prevented an attack, then it also detected the attack and could alert the operator in some useful way while also blocking it. Likewise, advanced technology made it possible to first detect an attack and then protect against it in either a static or dynamic fashion. One can imagine that the five core CND capabilities of protect, monitor, detect, analyze and respond could all exist together on one platform as technology continues to mature.

(U) IPSs are considered as the convergence of the fourth generation of firewall and IDS technologies. IPSs can monitor traffic and decide whether to drop or allow traffic based on expert analysis. These devices normally work at different areas in the network and can proactively monitor suspicious activity that may otherwise have bypassed the firewall. A complete network IPS solution has the ability to enforce traditional static firewall rules and administrator-defined whitelists and blacklists.

(U) The two main types of IPSs are:

- (U) Host-Based IPS – runs software directly on workstations and servers, and can detect and prevent threats aimed at the local host
- (U) Network-Based IPS – monitors from a network segment level, and can detect and prevent both internal and external attacks

(U) Host-Based IPS (HIPS) - As with HIDS, HIPS relies on agents that are installed directly on the system being protected and are closely bound to the operating system kernel and services. This allows system calls to the kernel or APIs to be monitored and intercepted in order to prevent and log attacks. In addition, data streams and the environment that are specific to a particular application may be monitored in order to protect against generic attacks for which no signature exists.

(U) Network-Based IPS (NIPS) - NIPS combines features of a standard IDS, an IPS, and a firewall. Packets appear at either the internal or external interface and are passed to the detection engine to determine if the packet poses a threat. Upon detection of a malicious packet, an alert is raised, the packet is discarded, and the flow is marked as bad. This results in the remaining packets of that particular TCP session arriving at the IPS device and immediately being discarded.

(U) In both types of IPSs, attack recognition is usually accomplished by known-attack detection or anomalous behavior detection.

2.6.3.6.2 (U) Usage Considerations

2.6.3.6.2.1 (U) Implementation Issues

(U) A number of challenges to implementing an IPS device stem from the inherent nature of being designed to work in-line, thus presenting a potential choke point and single point of failure. Performance of the network can be seriously impacted. Increased latency and reduced throughput could become problematic as IPS devices struggle to keep pace with high speed networks. A NIPS device must perform much like a network switch and meet stringent network performance and reliability requirements.

(U) Another potential problem deals with false positives. Although not as critical for an IDS device, false positives can be far more serious and far reaching for an in-line IPS device. The result can be a self-inflicted DoS condition.

2.6.3.6.2.2 (U) Advantages

(U) The basic advantage of an IPS in comparison to an IDS is the ability to not only detect attacks, but also to block them. An IPS acts to combine previous single-point security solutions (i.e., firewalls for access control and IDS for hackers) into a solitary architecture that is capable of blocking network attacks, intrusions, viruses, malicious code, and spam. For zero-day attacks where the virus is previously unknown, current IPS technologies can utilize databases of known protocol weaknesses and anomalous behavior techniques (also known as heuristics) to identify malicious traffic.

(U) The benefits of deterministic intrusion prevention can be summarized as:

- (U) Proactive protection from the network security infrastructure
- (U) Operational efficiencies due to reduced need to react to event logs for protection
- (U) Increased coverage against packet attacks and zero-day attacks

2.6.3.6.2.3 (U) Risks/Threats/Attacks

(U) Few barometers exist to provide an indication as to how much software or tools are needed to protect an organization's systems. Another risk is the false sense of confidence within an organization once IPS is deployed. Without adequate training of personnel and proper implementation and maintenance by service providers, the IPS remains at risk.

2.6.3.6.3 (U) Maturity

(U) As stated previously, the IPS technology is new and emerging. While IPSs represent a significant advancement over its predecessors, the IPS technology is just beginning to evolve and gain acceptance in industry. A recent trend of consolidation within the IPS industry has been observed, and shows no signs of slowing. The aim of these mergers is to acquire capabilities that can be re-branded into a resultant technology that is marketable as a new form of IPS.

(U//FOUO) The maturity of the various sub-technologies of the Intrusion Prevention System technology area is rated Early (TRL 1-3).

2.6.3.6.4 (U) Standards

(U) Due to the sensitive nature of most security events, the information will have to be sufficiently abstracted and shared via a standardized protocol. The current best candidate is the Intrusion Detection Message Exchange Format (IDMEF). Participating organizations would be able to exchange security-related information (i.e., new protocol anomaly patterns and worm outbreaks). Although a handful of collections have currently embarked on this endeavor, adoption in production systems is sparse thus far.

2.6.3.6.5 (U) Cost/Limitations

(U) As with any new emerging technology, IPS can be widely categorized as relatively expensive. Costs include the initial investment in IPS devices, as well as continual costs of IPS upgrades, maintenance, training, and management.

(U) One potential limitation of HIPS is that given the tight integration with the host operating system, future OS upgrades could cause problems.

2.6.3.6.6 (U) References

(U) “Intrusion Prevention: A White Paper,” www.nitroguard.com.

(U) “Intrusion Prevention Systems,” by Mike Barkett, July 2004, http://www.nfr.com/resource/downloads/SentivistIPS-WP.pdf.

(U) “Intrusion Prevention Systems (IPS),” January 2004, http://www.nss.co.uk/WhitePapers/intrusion_prevention_systems.htm.

(U) “Intrusion Prevention Systems (IPS): Next Generation Firewalls, A Spire Research Report – March 2004,” by Pete Lindstrom, http://www.toplayer.com/generic/Spire_IPS_Whitepaper.pdf.

2.6.3.7 (U) User Activity Profiling

2.6.3.7.1 (U) Technical Detail

(U//FOUO) There are many challenges in detecting and responding to insider misuse, including analyzing how insider misuse differs from penetrations and other forms of misuse by outsiders. The differences among users themselves vary by responsibility and involve either physical presence and/or logical presence. For example, there may be logical insiders who are physically outside and physical insiders who are logically outside. Technology solutions for user profiling are primarily focused on activity monitoring those insiders who are both logical and physical users.

(U//FOUO) User activity profiling attempts to learn normal behavior patterns with respect to key observed or derived features (i.e., resource usage and temporal patterns). There are different degrees of logical insiders related to the nature of the systems and networks involved, the extent to which authentication is enforced, and the exact environment in which a user is operating at the moment.

(U//FOUO) Profiling the behavior of not only individuals, but also the programs they operate can be a useful reference for detecting potential intrusions against systems. In general, anomaly detection techniques for profiling program behavior evolve from memorization to generalization using both host-base and network-based agent structures. These techniques often employ machine-learning techniques that can generalize from past observed behavior to the problem of intrusion detection.

2.6.3.7.2 (U) Usage Considerations

2.6.3.7.2.1 (U) Implementation Issues

(U//FOUO) Detecting insider misuse must rely heavily on user profiling of expected normal behavior as well as application-specific rules. The goal of monitoring programs is to be able to detect potential insider misuse by noting irregularities in user activities or program behavior and without extensive false alarms. These monitors often start from the development of a simple equality matching algorithm over time, and evolve to a feed-forward back-propagation neural network for learning program behavior, and finally to approaches for recognizing recurrent features in activity execution. In order to detect future malicious activities against systems, intrusion detection systems must be able to generalize from past observable behavior.

(U//FOUO) The validation of profiling systems is problematic, and usually relies on some variant of cross profiling, wherein fine-grained system measurements for one subject are played through the trained profile of another. Measurements can include network traffic activity, identity of current programs being executed, user typing speed, time of day, etc. Typical measures of detection effectiveness include time to detection and probability of detection for, say a user typing in a different way then normal or a window of unusual commands being issued. Unfortunately, this approach cannot be used to make strong claims about effectiveness against malicious use, but rather about discrimination between examples of use that are, to the best of the analyst’s knowledge, legitimate.

(U//FOUO) For large enterprise environments in which monitoring key strokes are not considered practical, some effort has been made to use triggers to initiate monitoring, plus monitor key-stroke dynamics. Triggers are most useful when closely monitored for false alarm control. Keystroke dynamics tend to be much less reliable in general, particularly when the differences in a typist’s frame of mind or the time of day must be considered.

(U//FOUO) Among the issues in implementing an activity monitor solution are providing a real-time database relating to physical whereabouts and extending statistical profiling to accommodate subtle computer usage variants. Further considerations should also take into account personal behavior such as intellectual and psychological attributes.

(U//FOUO) As an example of an intellectual attribute, consider writing styles. There are already a few tools for analyzing natural-language writing styles. Profiles of individual-specific ‘misspellings,’ the frequency of obscenities and the choice of explicit expletives, the relative use of obscure words, and measures of obfuscation proclivities and meanderings are also useful.

2.6.3.7.2.2 (U) Advantages

(U//FOUO) User profiling has long been used with some success to detect masquerader attacks and insider abuse. In general, profiling can be applied to any process under observation, such as the system call stream from programs and invites analogy to process control. The basic paradigm is to alert when the process under observation exhibits behavior that is extremely unusual with respect to learned norms.

(U//FOUO) As previously discussed, there are generally two types of intrusion detection systems: misuse detection and anomaly detection. The most significant advantage of misuse detection approaches is that known attacks can be detected fairly reliably and with a low false positive rate.

2.6.3.7.2.3 (U) Risks/Threats/Attacks

(U//FOUO) Masquerader and insider abuse pose fundamentally different problems than traditional detection solutions were intended to resolve. The masquerader may be detected by stylistic differences, while the insider can train his profile so that the eventual exploit appears normal. The difficulty is exacerbated by the problem of a hit and run attack, where the exploit is one event in an otherwise normal stream.

(U//FOUO) Another difficult to resolve problem is the false alarm. Even with fine-grained user profiles, user job functions mature and profiles change over time. There is a serious risk that a tool’s alarm generation capability will be greatly limited to reduce the number of false alarms being generated.

(U//FOUO) Many Government organizations strongly endorse the use of proprietary COTS IDS-like software that are unsecure, unreliable, and nonsurvivable. There are few emerging intrusion detection systems that are completely reliable at detecting hitherto unrecognized insider misuse. The reality that COTS intrusion-detection tools are not oriented toward insider attacks, unknown forms of misuse, intelligent results interpretation, and long-term evolution presents a very significant reason for closer evaluation of GOTS solutions.

2.6.3.7.3 (U) Maturity

(U//FOUO) Efforts to date have concentrated on relatively straightforward statistical measures, thresholds, weightings, and statistical aging of the profiles, independent of particular users. The basic problem with tools that automatically learn user models from things like what applications the person uses (order important) and the associated timing information is scalability and computation time. For this reason, current solutions are limited to enclave-level networks.

(U//FOUO) The maturity of the various sub-technologies of the User Activity Profiling technology area is rated Early (TRL 1-3).

2.6.3.7.4 (U) Standards

(U//FOUO) There are no standards that address this technology need. However, USSTRATCOM has published a CND Insider Threat Requirements document that addresses basic objectives and needs for insider threat technology solutions.

2.6.3.7.5 (U) Cost/Limitations

(U//FOUO) Simple activity monitor technologies that trigger more large-scale monitoring are not expensive to deploy. The cost is dependent upon the size of the operational system in which they are being placed and the maintenance and support cost to operate and manage them.

2.6.3.7.6 (U) Dependencies

(U//FOUO) Adequate controls of insider misuse suggest that better system security is necessary as one part of the solution. There is a fundamental need for better differential access controls (access control lists, compartmentalized protection, fine-grain roles, etc.). There is also a need for better user authentication to prevent intruders from gaining insider access and to provide positive identification of insiders that might diminish their ability to masquerade as other insiders and to otherwise hide their identities.

2.6.3.7.7 (U) Alternatives

(U//FOUO) Personal on-line behavior can also be profiled statistically by extending the analysis information that is recorded, such as with whom an individual tends to exchange e-mail, which Web sites are visited regularly, and even what level of sophistication the user appears to exhibit. This is only a minor extension of what can be done with monitor tools available today.

2.6.3.7.8 (U) Complementary Techniques

(U//FOUO) There are a few relative differences in detecting insider misuse compared with outsider-initiated misuse, but these differences do not seem to be intrinsic. Instead, the differences might involve the following:

- (U) Information to be gathered
- (U) Rules given to an expert system
- (U) Parameters used to tune the profile-based analysis
- (U) Priorities associated with different modes of misuse
- (U) Urgency accorded to various responses

(U//FOUO) Some new inference tools might be useful, but they could also be developed generally enough to be applicable to outsider misuse as well.

2.6.3.7.9 (U) References

(U) “CND Insider Threat Requirements, V0.43,” USSTRATCOM, August 2004.

(U) "The Challenges of Insider Misuse," by Peter G. Neumann, Sri Computer Science Lab, Post-Workshop Version, 23 August 1999, Prepared For The Workshop On Preventing, Detecting, And Responding To Malicious Insider Misuse, 16-18 August 1999, At Rand, Santa Monica, California.

(U) "Evaluating Software Sensors for Actively Profiling Windows 2000 Computer Users," by Jude Shavlik, University of Wisconsin-Madison, Mark Shavlik, Michael Fahland, Shavlik Technologies, St. Paul, Minnesota.

(U) "Learning Program Behavior Profiles for Intrusion Detection," by Anup K. Ghosh, Aaron Schwartzbard & Michael Schatz, Reliable Software Technologies Corporation.

2.6.3.8 (U) Cyber Attack Attribution

2.6.3.8.1 (U) Technical Detail

(U//FOUO) Approaches in defending network-based intrusions are categorized as intrusion prevention, intrusion detection, intrusion tolerance, and intrusion response. Response mechanisms usually take two approaches: localizing the source of the attack using traceback techniques or reducing the intensity of the attack by blocking attack packets.

(U//FOUO) To hide their identity, network-based intruders seldom attack directly from their own hosts, but rather from hosts acting as intermediate stepping-stones or zombies. Spoofing the return address in a one-way communications is also a common practice. In order to identify the intruder behind these stepping-stones, it is necessary to be able to trace through each intermediate host and construct the correct intrusion connection chain. Traceback is the term used to describe the technology for reconstructing the connection chain to the original IP host.

(U//FOUO) There are several different approaches of tracking and tracing attacks via route-based distributed packet filtering, some of which include:

- (U) Hop-by-Hop Traceback
- (U) Backscatter Traceback
- (U) CenterTrack
- (U) ICMP Traceback or iTrace
- (U) Hash-Based IP Traceback

2.6.3.8.1.1 (U) Hop-by-Hop Traceback

(U) The most common and basic in use today, hop-by-hop traceback traces large, continuous packet flows that are currently in progress and that originate from spoofed source addresses (i.e., DoS packet flood attack). Starting with the Internet Service Provider's (ISP's) router closest to the victim, an ISP administrator uses the diagnostic, debugging, or logging features of the router to characterize the nature of the traffic and determine the input link of the attack. The administrator then moves to the upstream router where the attack packets are coming from. This diagnostic procedure and trace backwards is repeated—hop-by-hop—until the source of the attack is ultimately found.

2.6.3.8.1.2 (U) Backscatter Traceback

(U) The backscatter traceback technique makes clever use of the large number of invalid source addresses that are characteristic of contemporary distributed denial-of-service (DDoS) attacks. Once a DDoS attack has been identified, routers are configured by the ISP to reject all packets destined for the victim. This will result in a slew of destination unreachable error message packets or backscatter that can be routed for capture. This technique makes use of the fact that the Internet Address Naming Authority (IANA) has not allocated several large blocks of IP addresses for global routing.

2.6.3.8.1.3 (U) CenterTrack

(U) The CenterTrack approach improves traceability by adding an overlay network, or auxiliary network formed from the joining of new physical/logical connections on top of the existing one. The overlay network is optimized for hop-by-hop tracing and analysis because of having only a small number of hops between edge routers. Intended DoS flood attack packets can be diverted to the overlay network which is equipped with special-purpose tracking routers.

2.6.3.8.1.4 (U) ICMP Traceback or iTrace

(U) The fundamental concept is that about once in every 20,000 packets, a router sends an ICMP traceback message (called an iTrace packet) to the same destination address as the sampled packet (or to an outboard monitor). The destination (or monitor) collects and correlates the tracking information to successfully trace the attack.

2.6.3.8.1.5 (U) Hash-Based IP Traceback

(U) All of the traceback methods described so far have limited capability because each of these techniques requires a large amount of attack traffic to support tracking. Arguably the most promising new research approach, Hash-Based IP Traceback (also known as Single-Packet IP Traceback) offers the possibility of making feasible the traceback of single IP packets. The fundamental idea is to store highly compact representations of each packet rather than the full packets themselves. These compact representations are called packet digests and are created using mathematical functions called hash functions. Hash-based IP traceback uses a system known as Source Path Isolation Engine (SPIE).

(U//FOUO) Using a timing and marking approach, current research has been able to develop a partial solution to the traceback problem. The ARDA Footfall Project at North Carolina State University is currently evaluating a new method of embedding that works in real-time and spreads the delay across all the packet pairs selected. The method is based on actively watermarking the traffic timing so that traffic can be correlated across stepping stones, or intermediate hosts, and through the network. The basic idea is to manipulate the timing in such a way that the traffic is uniquely recognizable by an analysis program. Watermarking techniques create a method of traceback that is almost arbitrarily robust to attempts by attackers to perturb traffic timing to avoid traceability. It is expected that the approach developed through the Footfall Project will be the first deployable partial solution on DoD networks. This technology transition should take place by the end of 2005. Therefore, the integration of a partial traceback solution on the DoD network will take place before GIG Technology increment 1.

2.6.3.8.2 (U) Usage Considerations

2.6.3.8.2.1 (U) Implementation Issues

(U) Route-based traceback is a very labor-intensive, technical process and often requires cooperation among bordering ISPs to complete the trace. Routers at each hop will need sufficient diagnostic capabilities to follow the trace. In addition, as in tracing a phone call by the police, the attack must remain in progress in order for the trace to be completed back to its origin.

(U//FOUO) There are also policy implications that need to be considered. Careful coordination needs to be in place as attacks can flow across administrative, jurisdictional, and national boundaries. Unlike passive defense techniques, active traceback can involve privacy laws. These laws directly impact automated systems that perform investigations for law enforcement. While the legal issues prevent Government use of some available commercial systems, private firms use them to gather information or actively react to network intrusions.

(U//FOUO) The three U.S. Laws that dictate legal considerations are the Electronic Communications Privacy Act, ECPA (18USC2701), the Wiretap Act (18USC2511), and the Trap and Trace Act (18USC3121). Since these laws were not written directly to protect against computer network crime, additional case law and interpretation is necessary to determine their exact relationship to traceback.

(U//FOUO) There are also some less defined areas within the statutes themselves. Specifically, techniques that involve some form of content monitoring or fingerprinting may violate privacy issues. Privacy protects the original packet contents, not a digested metric of the packet itself. Additionally, there are distinctions made between collecting and disclosing to others, voluntary versus non-voluntary collection, and stored access versus real-time access. The bottom line is that a traceback technology solution could violate the law under some conditions.

(U//FOUO) Unlike actual adversaries, legal restrictions and the rights of U.S. citizens limit the capabilities of Defensive Information Operations (DIO) services. For example, a Red Team cannot target public domain servers being used as avenues to place malicious code on DoD hosts. However, real adversaries do target and exploit public domain servers at will. Also, even if all legal restraints were lifted, robust tools were developed, and additional defensive resources were available, the ability to respond to attacks would still be challenged by political considerations based on adversarial relationships.

2.6.3.8.2.2 (U) Advantages

(U) Backscatter traceback is a fast and efficient method of countering DDoS flood attacks.

(U) An advantage of the CenterTrack approach is that special-purpose tracking and analysis features are not needed on all routers, but only on the edge routers and those for special-purpose tracking.

(U) All of the probabilistic traceback approaches depend on auditing very sparse samples of large packet flows and thus are well suited for attacks that generate massive packet flows, such as DDoS floods.

2.6.3.8.2.3 (U) Risks/Threats/Attacks

(U) Hop-by-hop traceback of DDoS attacks can be adversely affected due to resource consumption of bandwidth and processing power in the network by the DDoS attack itself.

(U) Backscatter traceback is heavily dependent upon specific characteristics of DDoS attacks it was defined to defeat. Like many other approaches designed to work against DDoS flood attacks, its success depends on a large number of attack packets being directed to a victim and is therefore, not as effective to subtle attacks. As attack methodology continues to advance (i.e., DDoS attack tool that uses randomly selected IP address from valid IANA allocation), the backscatter traceback technique will eventually be defeated. In addition, attacks that do not forge zombie source addresses would also be able to defeat this technique.

(U) The CenterTrack approach fails when an attack originates inside an ISP’s network. In addition, high scalability is uncertain for DDoS attacks with many entry points into the ISP’s network.

(U) The iTrace approach can be defeated or disrupted by sending spoofed iTrace packets. Therefore, iTrace packets must include an authentication field.

2.6.3.8.3 (U) Maturity

(U//FOUO) DoD organizations investigating attacks currently use manual techniques. There is no current automated solution to traceback. Existing approaches have focused on identifying the set of correlated connections in the connection chain. These approaches have overlooked the serialization of those correlated connections, thus providing an incomplete solution (Wang, March 2004).

(U//FOUO) The maturity of the various sub-technologies of the Cyber Attack Attribution technology area is rated Early (TRL 1-3).

2.6.3.8.4 (U) Standards

(U//FOUO) One emerging standard that will help—but not solve the traceback problem—is implementation of the IPv6 protocol. Another standard that could significantly reduce the problem would be requiring all routers to place their own unique ID in the protocol of each packet they receive. The drawback to this approach is that the routing overhead would increase greatly, and all existing hardware would need to be replaced. One technique that uses a query approach between routers employs the Intrusion Detection and Isolation Protocol (IDIP) developed through a Defense Advanced Research Projects Agency (DARPA) project. At this time no standard is being promoted for resolving the traceback gap area.

2.6.3.8.5 (U) Cost/Limitations

(U//FOUO) Route-based distributed packet filtering for attack prevention and traceback, has been widely studied. Tracing IP packets with forged source addresses requires complex and often expensive techniques to observe the traffic at routers and reconstruct a packet’s real path traveled. Tracing becomes ineffective when the volume of attack traffic is small or the attack is distributed.

(U//FOUO) Currently available Traceback tools that can be used by DoD are primarily Government-off-the-Shelf (GOTS). Additionally, there are a limited number of authorized organizations that can use these tools.

(U) Policy implications can limit the tracing of attacks that go beyond administrative, jurisdictional, and international boundaries and will most likely depend upon the trustworthiness, cooperation, and skill of other ISPs.

(U) For the CenterTrack approach, an increase in the overall complexity can result in operational errors (i.e., routing updates). Also, the overhead inherent in creating IP tunnels could amplify a DoS flood's negative effects on the network.

2.6.3.8.6 (U) Dependencies

(U) International agreements will need to be established in order to formalize the cooperation needed to make the techniques effective. This may need to include agreements to share traceback technology if the overall level of skill needed to complete a trace is not sufficient.

2.6.3.8.7 (U) Alternatives

(U//FOUO) Within DoD, the alternatives to traceback using traditional techniques form the basis of the currently deployed Defense-in-Depth approach. Until deployable automated traceback can be developed, only defensive approaches and manual techniques are available.

2.6.3.8.8 (U) Complementary Techniques

(U) Other research works such as various intrusion detection models, data mining-based models, and IDSs are complementary to the aforementioned traceback techniques.

2.6.3.8.9 (U) References

(U) "Advanced and Authenticated Marking Schemes for IP Traceback," by Dawn Song and Adrian Perrig, University of California Berkeley, DARPA Research Project N6601-99-28913.

(U) The Footfall Project, http://footfall.csc.ncsu.edu/index.htm.

(U) "A Little Background on Trace Back," CSC 774 Network Security, Spring 2003, http://discovery.csc.ncsu.edu/~pning/Courses/csc774/on-trace-back.pdf.

(U) "The Loop Fallacy and Serialization in Tracing Intrusion Connections through Stepping Stones," by Xinyuan Wang, North Carolina State University, SAC' 04, March 14-17, 2004, Nicosia, Cypress.

(U) "Technical, Legal, and Societal Challenges to Automated Attack Traceback," by Susan Lee and Clay Shields, Technical, ITPro, May/June 2002.

(U) "Tracking and Tracing Cyber Attacks: Technical Challenges and Global Policy Issues," by Howard F. Lipson, CMU/SEI-2002-SR-009, November 2002, http://www.cert.org/archive/pdf/02sr009.pdf.

2.6.3.9 (U) Correlation Technologies

2.6.3.9.1 (U) Technical Detail

(U//FOUO) Correlation technologies are tools that provide the capabilities to perform data aggregation, correlation, reduction, and analysis. With the widespread integration of security solutions such as intrusion detection and protection/prevention systems into the global networked environment, comes an increased need to implement tools that provide for the management of the data collected by these systems.

(U//FOUO) Many security solutions generate enormous quantities of data. It has become necessary to use applications to perform the comprehensive analysis necessary to correlate security event data in a timely (real-time/near real-time) manner. The analysis of this data allows for the identification of the anomalies and trends that are buried within the data. These events must then be displayed and reported in the most comprehensive method possible in order to respond immediately to an event.

(U//FOUO) The GIG architecture calls for a significant increase in network bandwidth throughout the entire system from the core to the remote wireless endpoints. As network bandwidth increases, the job of CND becomes more challenging. Both the volume of packets inspected by CND technologies and the number of alerts generated by the CND tools increase tremendously. For this reason alert correlation becomes increasingly important through each of the GIG IA increments.

(U) As stated by Haines et al:

- (U) [Correlation] systems take as input the output produced by low-level sensors such as intrusion detection systems, firewalls, and integrity checkers. Correlators issue reports that group together related alerts and events to provide an improved understanding of a suspected cyber attack and to help analysts identify and dismiss false alarms. Human administrators use these reports to understand the state of their network and select an appropriate response

- (U) The goal of correlation is to provide high-level reasoning beyond low-level sensor capabilities

(U//FOUO) As a data analysis tool, the correlation tool pulls attack, reconnaissance, and log data from a number of sources (e.g., network and computer sensors, NIDS, HIDS, firewalls, packet filtering routers, and vulnerability assessment tools). It also normalizes data from stovepipe systems, correlates, prioritizes, and reduces that data. Using the normalized data, the tool generates graphical representations of data and generates reports. The normalized data can then be used later for forensics analysis. The data presented in the reports would trigger the active response capability to provide immediate mitigation to a highly destructive event.

(U//FOUO) In the current state of the art, security vulnerability analysis tools consider individual vulnerabilities independent of one another. Moreover, they analyze single machines only, in isolation from other machines in the network. However, the interdependency of vulnerabilities and the connectivity of a network make such analysis incomplete. While a single vulnerability itself may not pose a significant direct threat to a system, a combination of vulnerabilities may. Thus even well administered networks are vulnerable to attacks, because of the security ramifications of offering a variety of combined services. That is, services that are secure when offered in isolation nonetheless render the network insecure when offered simultaneously.

(U//FOUO) Many current tools address vulnerabilities in isolation and in the context of a single host only. This can be extended by searching for sequences of interdependent vulnerabilities, distributed among the various hosts in a network. This approach is called Topological Vulnerability Analysis (TVA).

(U) Correlation tools include components to perform data capture (agent), data collection and storage (manager), organization and tagging (database), and a user interface (console or web-based). The data being manipulated by the system internally should be encrypted.

2.6.3.9.2 (U) Usage Considerations

2.6.3.9.2.1 (U) Implementation Issues

(U) As correlation technologies are currently in the research and development stage, implementation issues have not yet been fully explored. It is expected, however, that there will be some significant obstacles that must be addressed. For example, some correlation approaches rely on the sensor’s ability to learn what normal network traffic is, and thus develop the ability to identify and correlate unusual events. If the correlation engine requires knowledge of typical adversary behavior, this too must be analyzed, tailored to the specific network segment, and incorporated into the system. If the correlation engine requires knowledge of the network architecture or vulnerabilities, the capability to readily include this information, preferably in a mostly automated manner, must be integrated.

(U) Intrusion detection on an encrypted network in itself presents significant challenges that must be addressed before the next step of correlation can be taken.

(U) Implementing a collective set of correlation technologies, rather than a single one, to further enhance analysis capabilities has significant cost, integration, maintenance, and management implications.

2.6.3.9.2.2 (U) Advantages

(U) The advantage to correlating alert information, as opposed to having teams of analysts digging through voluminous near-raw alert data, is significant as the bandwidth of the GIG increases. It will not be practical to rely on pure human analysis of this data in the future. It is critical to CND to have the ability to reduce the overall volume of alert information, as well as correlate similar alerts, disparate alerts, alerts detected by a variety of sensor systems, and alerts collected on a variety of different network systems. It will be important to be able to correlate alerts across different tiers within the GIG architecture. It will be critical to have this information available to the key decision makers at all levels within the GIG in near-real time. And, eventually, the ability to include mission priorities in the correlation process will put the CND analyst in a position to be proactive about protecting the mission rather than reactive.

(U) With the assistance of correlation technologies, the analyst is better able to quickly assess a current status of the network by focusing on manageable information sets. With the assistance of advanced visualization tools, this process is further enhanced. From this information, decisions on response actions can be made and implemented. For future iterations of correlation capabilities, it is desirable to overlay mission priorities on the correlation analysis to see if the mission is targeted or impacted as a result of a malicious network event, or if response actions will impact the mission in an undesirable manner.

2.6.3.9.2.3 (U) Risks/Threats/Attacks

(U) There are three key risks:

- (U) The first is the user’s ability to trust that the data has been correlated accurately
- (U) The second is the ability to trust that the correlation process has not dropped key alerts
- (U) The third to the ability to trust that the correlation process has not developed false positives

(U) The only way to address these risks is to continue to invest in correlation research and development to improve these systems.

(U) It is conceivable that an adversary could try to distract a correlation system by intentionally triggering alerts and hiding the real attack traffic in the subsequent smokescreen. This is something to be addressed by the research community.

2.6.3.9.3 (U) Maturity

(U) As previously stated, correlation technologies are currently in the research prototype stage. There are no advanced correlation technologies available off-the-shelf today. While some COTS sensors have limited data reduction capabilities, such as reducing the individual alerts due to a scan, true analytical correlation with disparate alerts is not commercially available. However, alert correlation has been the subject of recent research with proof of concept technologies currently being explored showing promising results. Several of these are referenced below.

(U) Research has shown that the combination of different correlation technologies, rather than a single technology, can yield even better results. This allows the disparate systems to focus on their strengths, and compensate for one another's weaknesses.

(U//FOUO) The maturity of the various sub-technologies of the Correlation technology area is rated Early (TRL 1-3).

2.6.3.9.4 (U) Standards

(U) There are no correlation standards at this time.

2.6.3.9.5 (U) Cost/Limitations

(U) Correlation technology cost is unknown at this time. However, one can assume that it would cost in the range of an advanced IDS. Some advanced IDSs will include correlation capabilities. Costs associated with the manpower to monitor the systems can be a limitation depending on the number of sensors being managed/monitored per analyst and the volume of data collected.

2.6.3.9.6 (U) Dependencies

(U) Correlation systems rely in total on the alert information that it can access. It is absolutely essential for advanced accurate sensors to precede the implementation of correlation technologies. These sensors must be effective in detecting malicious activity on encrypted network segments.

(U) Correlation systems also depend on the ability to display the correlated results. While some systems can generate reports or visual aids, much work can be done to improve current prototypes. Ideally, correlation results would be fed into a complete situational awareness picture for further analysis.

2.6.3.9.7 (U) Alternatives

(U) The alternative to correlating alert information is to simply increase the overall assurance of a network and prevent attacks from the outset. Since this is clearly unrealistic, the remaining alternative is to rely on human analysis to draw the correlation relationships. This would be a significant challenge with every increasing bandwidth, and the sheer volume of network components that must be monitored.

2.6.3.9.8 (U) Complementary Techniques

(U) Again, human analysts can correlate information manually to some degree. However, these capabilities can be improved upon significantly with the proper use of computing, mathematical, and modeling power.

2.6.3.9.9 (U) References

(U) "Adaptive, Model-Based Monitoring for Cyber Attack Detection," by A. Valdes, K Skinner, Recent Advances in Intrusion Detection (RAID 2000), pp. 80-92.

(U) "A Mission-Impact-Based Approach to INFOSEC Alarm Correlation," by P. Porras, M. Fong, A. Valdes, Proceedings Recent Advances in Intrusion Detection, Zurich, Switzerland, October 2002, pp. 95-114.

(U) “Probabilistic Alert Correlation,” by A. Valdes, K. Skinner, Recent Advances in Intrusion Detection (RAID 2001).

(U) “Scyllarus Intrusion Detection Report Correlator and Analyzer,” by W. Heimerdinger, DARPA Information Survivability Conference and Exposition, Volume 2, April 2003, pp. 24-26.

(U) “The STAT Toolsuite,” by G. Vigna, M. Eckmann, R.A. Kemmerer, DARPA Information Survivability Conference and Exposition, Volume 2, April 2003, pp. 46-55.

(U) “Validation of Sensor Alert Correlators,” by J. Haines, D. Ryder, L. Tinnel, S. Taylor, IEEE Security and Privacy, January/February 2003, pp. 46-56.

(U) http://www.sdl.sri.com/programs/intrusion/.

(U) http://www.cs.ucsb.edu/~rsg/STAT/.

2.6.3.10 (U) CND Response Actions

2.6.3.10.1 (U) Technical Detail

(U//FOUO) U.S. Strategic Command (USSTRATCOM) defines CND Response Actions (CND RAs) as deliberate, authorized defensive measures or activities that protect and defend DoD computer systems and networks under attack or targeted for attack by adversary computer systems/networks. CND RAs extend DoD's layered defense-in-depth capabilities and increase DoD's ability to withstand adversary attacks.

(U) Response actions are taken as a result of a detected intrusion and can be either automated or manual—requiring a human in the loop to activate the response. Response actions are implemented to stop ongoing attacks, such as a denial-of-service, or to plug already exploited vulnerabilities from future network attack.

(U) Response actions can be construed as counter attack when the action reaches beyond the GIG controlled assets to target the source of the attack. There are currently notable legal limitations on such actions.

(U) Response actions can be proactive in nature, updating a security posture based on external intelligence or other sources or to prioritize mission critical asset protections prior to executing an operations plan. By the same token, proactive response actions can be targeted against adversary assets in support of an operation. This action generally falls under the computer network attack category and will not be discussed further herein.

2.6.3.10.2 (U) Usage Considerations

2.6.3.10.2.1 (U) Implementation Issues

(U) Clearly the ramifications of response actions can be far reaching, especially if the response does not take into consideration mission priorities. The actions must be well considered, and if there is time and opportunity, modeling the response in advance of implementing it can be advantageous. In cases where an active attack must be stopped, it will not be practical to take the time to do any modeling. In such an instance, an immediate short-term response can be taken, followed by a well-considered longer-term solution that has undergone analysis, and in some cases modeling.

(U) While automated response capabilities do exist in a limited capacity in some COTS and research prototype technologies, automated response is not currently a widely accepted practice. DoD policies and procedures limit or prohibit an automated response in most cases, and lack of experience and in-depth knowledge of CND capabilities makes the leadership chain hesitant to fully trust and use automated engines.

(U) When the technology becomes available, response actions need to be global solutions coordinated across multiple network enclaves, rather than localized implementations. There are bound to be significant conflicts resulting in temporary loss of mission critical assets otherwise.

(U) There is discussion between the CND community and the network management community as to who will actually implement the response actions, whether it is a CND analyst or a network management operator. As the GIG progresses, the lines between the two groups will continue to blur, and it will be absolutely critical for both to work hand-in-hand continuously. In many cases the technologies used to implement the responses will often be the same technologies that either detected or prevented some portion of the attack. It is impractical to think that a clean hand-off to the network management group will be possible. Response is also frequently an iterative process requiring a series of detected and analyzed intrusion detection alerts, followed by more and more refined response actions.

2.6.3.10.2.2 (U) Advantages

(U) Responding to a network attack provides the opportunity for the defenders to stop malicious network events and prevent the adversary from reaching its goals. Without implementing some sort of a responsive action, an adversary that has gained unauthorized access will have the luxury of time to collect further intelligence about the GIG network assets and see a wealth of sensitive data.

(U) The advantage of automated response is that malicious packets can be stopped within seconds of being detected. This packet race can be critical in blocking the adversary before more lethal network attacks are launched. It is not a perfect solution as the adversary will still be at least one packet ahead of the defenders, and this is particularly critical with the most sophisticated adversaries that have the one packet, one kill mentality. It is far better to prevent the attack in the first place than to have to monitor, detect, analyze, and then respond to unauthorized activity. The shorter the time window between detection and response, the closer one reaches prevention.

(U) The disadvantage to an automated response, however, is that the impact of the initial response may not be fully analyzed. This is why the two-tiered response approach provides additional value. Automated response must be resilient to adversary techniques intended to trigger it.

(U) Manual response, on the other hand, requires analysis time and human intervention, which can be slow and sometimes inaccurate. It does, however, allow for manual consideration of the mission impact and consultation with the appropriate chain of command.

2.6.3.10.2.3 (U) Risks/Threats/Attacks

(U) The risk of this technology is that response actions that are not well considered can have a detrimental impact on mission-critical GIG network functionality. Loss of functionality can be far reaching and result in significant down time to the user community. A negative impact of this sort could cause the user community and the chain of command to lack trust, and therefore not use the response technology, which would leave the networks vulnerable once again.

(U) If the adversary were able to trigger the response technology in some way to also make it untrustworthy, or to cause an analyst to disable the capability, there would be a negative impact on the GIG. In this case the technology would actually provide an additional control surface for the adversary to exploit—something which has been a point of interest in the risk assessment.

2.6.3.10.3 (U) Maturity

(U) There are available today a handful of CND technologies with integrated response capabilities. For example, a commercial DoS discovery technology is able to monitor, and analyze packets, and once a threshold has been crossed, alert the operator that a DoS has been detected. The technology then recommends a course of action to block the attack, which can be implemented either manually or automatically in a neighboring perimeter router.

(U) Response capabilities have been the subject of much research within the DoD, as noted in the references section below. Research prototypes have been developed, and they show much promise, especially when paired with sophisticated correlation systems. The science of launching sophisticated response actions would also benefit tremendously from the capability to include mission-critical network assets, plans, alternatives, and the notion of timing. Research into advancing response technologies beyond their present state should yield capabilities and technologies far greater than what is available today, for both the DoD and its adversaries.

(U//FOUO) The maturity of the various sub-technologies of the CND Response Actions technology area is rated Early (TRL 1-3).

2.6.3.10.4 (U) Standards

(U) There are no current standards for response actions. Any standards for response should be closely tied to those for intrusion detection.

2.6.3.10.5 (U) Cost/Limitations

(U) Response technologies may be integral to other CND technologies, so the cost of the technology product should be explored as a unit and is expected to be similar to that of other CND technologies. Research costs to develop response technologies, however, are expected to be significant.

(U) The usefulness of response technologies will be limited by the ability to centrally manage a set of devices, and the number of deployed DoD experts available to operate the systems and make critical and timely decisions involving response actions.

2.6.3.10.6 (U) Dependencies

(U) Response technologies are highly dependent on reliable intrusion detection data, which is in turn dependent upon monitoring and analysis capabilities. Without these, coordinated, sophisticated response actions will be unattainable. In addition, it will be important for the CND analyst to have access to reliable and comprehensive situational awareness data in near real time to make decisions and monitor the effects of response actions. This situational awareness data should include operational plans and prioritization of mission-critical assets in a time-based schedule.

2.6.3.10.7 (U) Alternatives

(U) The alternative to response technologies is a manual process of analyzing intrusion detection information and manually updating the security posture based on engineering judgment. This rudimentary approach will give the adversary the advantage of time. Equally important, the CND analyst responsible for reviewing the intrusion detection data will be more likely to experience fatigue, miss critical events, or make mistakes recommending and implementing response actions.

2.6.3.10.8 (U) Complementary Techniques

(U) The only complementary technique to response actions is to constantly evaluate and update the security posture of the GIG network devices as a result of perceived or known network threats.

2.6.3.10.9 (U) References

(U) “Autonomic Response to Distributed Denial of Service Attacks,” by D. Sterne, K. Djahandari, B. Wilson, B. Babson, D. Schnackenberg, H. Holliday, and T. Reid, Proceedings of the 4th International Symposium on Recent Advances in Intrusion Detection (RAID), October 2001, pp. 134-149.

(U) “Cooperative Intrusion Traceback and Response Architecture (CITRA),” by D. Schnackenberg, H. Holliday, R. Smith, K. Djahandari, and D. Sterne, DARPA Information Survivability Conference and Exposition II, Volume 1, June 2001, pp. 56-68.

(U) “Electronic Quarantine: An Automated Intruder Response Tool,” by P. Brutch, T. Brutch, and U. Pooch, Proceedings of the 1998 IEEE Information Survivability Workshop (ISW’98), October 1998.

(U) “SARA: Survivable Autonomic Response Architecture,” by S. Lewandowski, D. Van Hook, G. O’Leary, J. Haines, and L. Rossey, DARPA Information Survivability Conference and Exposition II, Volume 1, June 2001, pp. 77-88.

2.6.3.11 (U) Automated IAVA Patch Management

2.6.3.11.1 (U) Technical Detail

(U//FOUO) Until recently, patch management had always been a labor and time-intensive ordeal with little or no support tools. Patch management tools are now available that automate much of the process, including discovery of reported vulnerabilities and patches, scanning systems for vulnerabilities and configuration status, assisting in the analysis and decision making process to decide which patches to deploy and when, testing proposed patches in controlled environments, deploying patches to systems, and verifying successful patch deployments.

(U//FOUO) Since patch management only addresses software defects that lead to vulnerabilities, management tools are being integrated into security and vulnerability management tools that can provide a more complete system management capability. These newer tools reduce the amount of human intervention now required with current solutions.

2.6.3.11.2 (U) Usage Considerations

2.6.3.11.2.1 (U) Implementation Issues

(U//FOUO) Best practices in patch management indicate that a thorough analysis of proposed patches must be conducted to assess whether the patch even applies, and if so, to what systems within the production environment. The potential impacts to those systems must be clearly understood and evaluated and a priority assigned to mitigating the vulnerability.

(U//FOUO) Vulnerabilities in widely used applications, such as Microsoft's Internet Explorer (IE), would have high priority because of the number of users, the pervasive use of IE by other applications, and the severity of the attacks that could be mounted against it. IE is one of those applications where extensive testing must be performed to understand the impact of the patch in the production environment. Fixing one security vulnerability problem could cause others to arise or could cause some functions of IE to stop working.

(U//FOUO) Patches must be implemented quickly to thwart attacks using discovered vulnerabilities. However, deploying untested patches in a production environment may prove more costly than the attack. All patches should be thoroughly tested before deployment on as many of the release configurations as possible. A patch is just that—a quick fix to correct a functional bug or to counter a security vulnerability. It is not uncommon for a patch that corrects one problem to cause one or more other problems.

(U//FOUO) Standard software releases should be periodically re-baselined to avoid patches colliding with each other and to simplify maintaining patch and configuration status.

2.6.3.11.2.2 (U) Advantages

(U//FOUO) Patch management technologies enable the automation of much of the labor-intensive aspects of identifying, analyzing, and deploying patches. As the complexity of network systems continues to grow, manually-based patch management techniques quickly demonstrated their inability to scale with it. Stand-alone patch management products answer the immediate need of businesses to provide some relief in mitigating vulnerabilities. Patch management capabilities are being integrated into vulnerability management and system management tools to provide security and administration personnel even more automated capabilities.

2.6.3.11.2.3 (U) Risks/Threats/Attacks

(U//FOUO) Patch management by itself is not a complete security solution. It only addresses software defects. It needs to be integrated into a system management capability that includes asset inventory, vulnerability, configuration, and policy management. According to the vulnerabilities reported from CERT (http://www.cert.org/stats/), the number of vulnerabilities that must be addressed by the patch management task has steadily increased through 2002 and is only slightly tapering off as indicated in Figure 2.6-3: (U) Vulnerabilities Reported from CERT. The total vulnerabilities reported (1995-2Q 2004): 14,686.

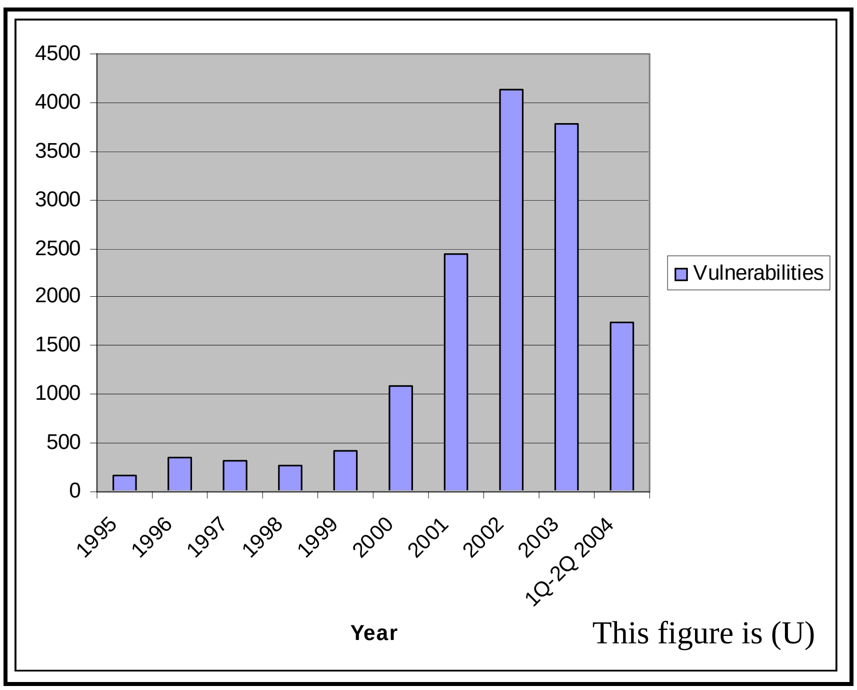


Figure 2.6-3: (U) Vulnerabilities Reported from CERT

2.6.3.11.3 (U) Maturity

(U//FOUO) The maturity of the various sub-technologies of the Automated IAVA Patch Management technology area is rated as Emerging (TRL 4-6).

(U) The maturity of patch management systems can be seen in the wide variety of products that are currently available. The following are examples of point solution products:

- (U) BigFix - BigFix Enterprise – http://www.bigfix.com
- (U) Ecora - Ecora Patch Manager – http://www.ecora.com
- (U) PatchLink Corporation - PatchLink Update – http://www.patchlink.com
- (U) SecurityProfiling - SysUpdate – http://www.securityprofiling.com/
- (U) Shavlik - Shavlik HFNetChkPro – http://www.shavlik.com
- (U) St. Bernard Software - UpdateEXPERT – http://www.stbernard.com
- (U) Microsoft – Software Update Services – http://www.microsoft.com

(U) The following are examples of security management products:

- (U) Citadel Security Software – http://www.citadel.com/
- (U) Configuresoft – Enterprise Configuration Manager – http://www.configuresoft.com

(U) The following are examples of security configuration management products:

- (U) Altiris – Client Management Suite – http://www.altiris.com/products/clientmgmt/
- (U) LANDesk Software – LANDesk Management Suite – http://www.landesk.com
- (U) ManageSoft – Security Patch Management – http://www.managesoft.com/solution/patchmanagement/index.xml
- (U) HP – Novadigm – http://www.novadigm.com
- (U) Novell (partner with PatchLink) – ZENworks Patch Management – http://www.novell.com/products/zenworks/patchmanagement/
- (U) Symantec/ON Technology (partner with Shavlik) – iPatch and iCommand – http://www.on.com

(U) The following is an example of a vulnerability management product:

- (U) Harris Corporation – STAT Scanner – http://www.stat.harris.com/index.asp

2.6.3.11.4 (U) Standards

(U//FOUO) There are no standards on patch management. Generally, all of the products offer similar capabilities following a de-facto industry best practice.

2.6.3.11.5 (U) Cost/Limitations

(U//FOUO) A variety of options exist for acquiring patch management products and services. Generally, there is a per seat price with break points at various quantities or an option to acquire an enterprise-wide license. Most vendors also offer a managed service capability.

2.6.3.11.6 (U) Dependencies

(U//FOUO) Patch management systems receive vulnerability and patch information from a number of industry and Government sources. Continued on-line access to these systems is required in order to maintain the most current information about patches.

(U//FOUO) An asset inventory of PCs and servers must be established and maintained that includes an up to date listing of operating system and applications with current patch and service pack status. The patch management system must periodically scan the PCs and servers to determine if there have been any changes to the status of the information on file. This status information is used during the analysis of a newly discovered patch or security vulnerability to determine which system may be vulnerable, what the likely impact will be to the enterprise, and what priority should be given to the mitigation of the vulnerability.

2.6.3.11.7 (U) Alternatives

(U) Basically, there are two types of patch management architectures available:

- (U) Agent-less: Agent-less based approach does not require any special software on the target machines. This approach typically uses RPC calls to scan machines for status and to deliver patches. This approach may result in some machines that cannot use such IT management tools to be patched manually.
- (U) Agent-based: Agent-based approach use special software delivered to each target system to enable communication with the patch server and to perform operations locally on the targeted machine. This approach typically uses TCP/IP to communicate with the server and could enable security features such as encryption that may not otherwise be available. Devices with limited bandwidth may require the use of agent-based software. Fortunately, vendors are making applications that support both capabilities.

(U//FOUO) Patch management systems are evolving to become an integral part of system management and vulnerability management applications. A separate patch management capability may not be needed in the near future.

2.6.3.11.8 (U) Complementary Techniques

(U//FOUO) Patch management is not a new concept. It is the evolution from a manual discovery and mitigation process to partially automated steps, and from discrete patch management tools to integrated security management tools. These tools include asset management, vulnerability assessment and management, policy compliance, configuration management, and patch management.

2.6.3.11.9 (U) References

(U) "Get Ready to Patch," by Foley and Hulme, InformationWeek, 30 August 2004, http://www.informationweek.com/story/showArticle.jhtml?articleID=45400083.

(U) "Patch Management is a Fast Growing Market," by Schroder, Colville, and Nicolett, Gartner, 30 May 2003, http://download.microsoft.com/download/a/2/6/a2625228-9394-4388-8dcf-de876ccfa88c/Gartner_patch_mgt_fast_growing.pdf.

(U) “Patch Management Vendor Overview,” by Mark Nicolett, Colville, and Silver, Gartner, 27 May 2004.

(U) “Patching Things Up, Emerging Technology,” CIO Magazine, 1 August 2003, http://www.cio.com/archive/080103/et_article.html.

(U) “Practical Patch Management,” NetworkWorldFusion, 21 October 2002, http://www.nwfusion.com/supp/security2/patch.html.

(U) “Robust Patch Management Requires Specific Capabilities,” by Nicolett and Colville, Gartner, 18 March 2004, http://www.knowledgestorm.com/sol_summary_63613.asp.

(U) “Security Patches: Plugging the Leaks in the Dike,” by Karen Krebsbach, Bank Technology News, August 2004, http://www.banktechnews.com/article.html?id=20040802B1GHC6AT.

(U) “SQL Slammer Lesson: Patch Management Is Not Enough,” by Mark Nicolett and John Pescatore, Tech Republic, 2 July 2003, http://techrepublic.com.com/5102-6264-5054273.html.

(U) “Taking Control of Vulnerabilities, Citadel Security Software Interview with John Pescatore,” Gartner Research, Interview conducted 27 April 2004, http://mediaproducts.gartner.com/gc/webletter/citadel/issue3/gartner1.html.

(U) The Need for Patch Management, Symantec, June 2004, http://sea.symantec.com/content/displaypdf.cfm?pdfid=29.

(U) “The Power of Optional Agent Arcitecture: Advantages of Managing Patches Remotely with UpdateEXPERT,” St. Bernard Software, Inc., 28 July 2003, http://www.stbernard.com/products/docs/OptionalAgent.pdf.

(U) “Vulnerability and IT Security Management Are Converging,” by Mark Nicolett, Gartner, 10 February 2004.

(U) “Vulnerability Management Technology Landscape,” by Mark Nicolett, Gartner, 11 September 2003.

2.6.4 (U) Network Defense and Situational Awareness: Gap Analysis

(U//FOUO Table 2.6-2 is a matrix of Network Defense and Situational Awareness technologies described in previous sections. The adequacy matrix is based upon 2008 capabilities.

Table 2.6-2:(U) Network Defense & Situational Awareness Technology Gap Assessment

This table is (U)

Technology Categories

IA Attributes			Firewalls (Host)	Firewalls & Filtering Routers (Network)	Virus Protection	Automated Patch Management	Honeypots & Honeynets	Situational Awareness	Vulnerability Scanning	Host-Based IDS	Network-Based IDS	Host-Based IPS	Network-Based IPS	User Activity Profiling	Attack Attribution	Alert Correlation	CND Response	Required Capability (RCD attribute)
IA Attributes	Protect	Vulnerability Assessment & Reporting	N/A	N/A	N/A	N/A	N/A	N/A		N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	IACND6, IACND7
	Protect	Unauthorized/ Malicious Activity Prevention					N/A	N/A	N/A	N/A	N/A			N/A	N/A	N/A	N/A	
	Protect	Configuration Change					N/A	N/A	N/A	N/A	N/A			N/A	N/A	N/A	N/A	IACND8, IACND9
	Monitor	Information Monitoring				N/A											N/A	IACND10
	Monitor	Information Presentation				N/A			N/A						N/A		N/A	IACND11
	Detect	Unauthorized/ Malicious Activity Identification				N/A			N/A									IACND12
	Detect	Unauthorized/ Malicious Activity Reporting				N/A			N/A								N/A	IACND13
	Analyze	Data Reduction & Correlation				N/A	N/A		N/A									IACND14

This table is (U)

Technology Categories

			Firewalls (Host)	Firewalls & Filtering Routers (Network)	Virus Protection	Automated Patch Management	Honeypots & Honeynets	Situational Awareness	Vulnerability Scanning	Host-Based IDS	Network-Based IDS	Host-Based IPS	Network-Based IPS	User Activity Profiling	Attack Attribution	Alert Correlation	CND Response	Required Capability (RCD attribute)
		Unauthorized/ Malicious Activity Analysis	N/A	N/A		N/A	N/A		N/A									IACND15
		Information Visualization & Sharing	N/A	N/A		N/A	N/A		N/A									IACND17
		Development & Coordination of COAs	N/A	N/A	N/A	N/A	N/A		N/A									IACND16, IACND18, IACND20
		Modeling & Simulation of COAs	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A			IACND19
	Respond	Response Actions			N/A		N/A		N/A					N/A				IACND21, IACND23
		Recovery Actions	N/A	N/A	N/A		N/A		N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	IACND22, IACND24

This table is (U)

11549

2.6.5 (U) Network Defense and Situational Awareness: Recommendations and Timelines

(U//FOUO) The following recommendations have been identified in the Network Defense and Situational Awareness Enabler. Without these, the GIG Vision cannot be fully satisfied. The recommendations are organized in the following categories: Standards, Technology, and Infrastructure.

2.6.5.1 (U) Standards

(U) One or more standards on sensor data are needed to address:

- (U) Format of sensor data
- (U) Semantics of sensor data

2.6.5.2 (U) Technology

(U) It is unlikely that today's protect technologies alone can stop sophisticated stealthy attacks. In order to raise the bar on the sophisticated risk-averse adversary, tomorrow's protect technologies must include capabilities such as:

- (U) Dynamic protection mechanisms capable of modifying the defensive structure either on-the-fly as a result of an adverse event or in a proactive organized defensive manner
- (U) Adaptive, self-learning capabilities that do not rely on previously known attack signatures
- (U) Ability to successfully protect encrypted network segments. As current protect technologies are not designed to operate on encrypted network segments, additional research and development is needed to develop new capabilities and technologies designed for such an environment.

(U//FOUO) In general, the Situational Awareness technologies represented by the current capabilities are not scalable to the needs of the GIG. More robust tools are needed to automatically collect and correlate a variety of information sources and to augment many of the I&W tasks that are now extremely manpower intensive. Additional processing requiring automation is the assessment of changes in an adversary's posture and perceived threat intent for all three levels of the Defense-in-Depth security strategy: computing environment, enclave, and network.

(U//FOUO) The scalability issue with current correlation tools, the need for collection capabilities, both at the packet level and from metadata sources on a very large enterprise, and the need to integrate some form of risk analysis based on current conditions has created several technology gap areas. These technology areas are currently being researched, and solutions are expected within the GIG Increment 1 time period.

(U//FOUO) Table 2.6-3 summarizes needs, gaps, and areas for exploration for Situational Awareness.

Table 2.6-3: (U//FOUO) Summary of Technology Gaps

This table is (U//FOUO)

Need	Gaps	Areas for Exploration
Develop and present the situation (via GUI). (The GUI supports all of the other needs listed below.)	3-D scientific data visualization tools for this application need enhancing.	Ways to effectively present to the user the security configuration and status of the enterprise.
	Interactive GUI tools and forms need developing specific to this application	The GUI needs to allow the user to respond to events. Management events would include changes in the network and new requirements; Operational events would include alerts, problems, and failures.
Application (high-level) security management and operations	Application-specific software tools need to be written for this DoD problem domain.	Managing changes in software to support changes in policy, developing CND COAs, deploying new CND services, and upgrading IAVM processes.
		Operationally performing the IAVM process, setting INFOCON levels dynamically, coordinating cyber awareness and reactions with other organizations.
Infrastructure (medium-level) security management and operations	Research products (e.g. Outpost, Network Policy Product) from Federally Funded Research and Development Centers (FFRDCs) need to be extended.	Managing security of web portals and servers, access lists in routers, database servers, modem pools, and policy settings in proxy servers.
		Operationally changing routing paths, accessibility to domain name servers, and the accessibility status of a modem port.
Security device (low-level) management and operations	Many COTS Intrusion Detection Systems (IDSs) exist. Applying them to large-scale DoD enterprise systems is a challenge.	Managing external-threat intrusion detectors, internal-threat sensors, and policy settings in firewalls.
		Operationally analyzing firewall logs, monitoring connections to the proxy server, and analyzing intrusion detection alerts.

This table is (U//FOUO)

(U//FOUO) In the area of enterprise-wide mapping of services/applications, advanced infrastructures require the mapper to manage, process, and interpret the volumes of data required to protect an information infrastructure. This includes strategies for discovery, data storage and retrieval, and visualization techniques to identify both network components and the defense posture they represent.

(U//FOUO) With the current passive mapping solution implemented on a portion of the DoD enterprise to meet the above needs, further implementation of the technology across the entire enterprise would provide a comprehensive solution. However, the focus of new research and tool development for enterprise-wide network monitoring and vulnerability assessments should take into account advances in intelligent agents that can potentially solve the problems faced with large-scale network situational awareness and defense posture discovery. The following gap areas need further research:

- (U//FOUO) Validate configuration management compliance of all network resources
- (U//FOUO) Validate INFOCON implementation conditions by combining with visualization and risk-based predictive tools
- (U//FOUO) Verify Ports and protocol adjudication and adherence
- (U//FOUO) Produce SA analysis and assessment tools using agent-based approaches that will allow the combination of mapping technologies

(U//FOUO) There is a basic gap in host systems and networks between what kinds of system uses are intended and what uses are actually specified or allowable based on installed applications. Application-based anomaly detection work has been effective at detecting novel threats against Internet servers. Anomaly detection approaches detect changes in the normal behavioral profile of the process and flag warnings of possibly corrupted processes. Anomaly detection systems trained to look at inside activity are now being viewed as having potential application to the insider threat technology solution. However, greater emphasis needs to be focused on detecting unknown modes of misuse, rather than just focusing so heavily on detecting known attacks. The existing statistical paradigms must be pursued and refined.

(U//FOUO) Reporting extremely unusual activity is important, but it is not enough. In addition, one promising approach is to describe classes of misuse probabilistically, so that much of the generalization potential of anomaly detection is retained but with improved sensitivity and specificity. Finally, signature detection is required for attacks manifest in single events or buried in a mostly normal stream (so that signal integration will not make it stand out sufficiently). We propose an innovative approach based on hybrid systems integrating anomaly detection (model-free inference) and Bayes (probabilistic, model-based).

- (U//FOUO) Use of zone/node sensors that operate on the concept of reporting status changes to their nearest neighbor
- (U) Geolocation of attacks

(U) A number of different automated approaches to the IP traceback problem have been suggested. However, no current method is completely effective in large-scale networks. This is known as the IP traceback technology gap.

- (U//FOUO) Performance/situational monitoring in the Black Core
- (U//FOUO) CND for high speed, high volume coalition services

- (U//FOUO) CND for high speed, high volume cross domain services
- (U) Automatically blocking DoD attacks

2.6.5.3 (U) Infrastructure

(U//FOUO) The creation and enterprise-wide implementation of an Enterprise Wide Sensor Grid (ESG) is essential to meet the needs of the UDOP. An ESG will provide collection capabilities for correlation and analysis of CND events and activities from single or multiple sensor categories (i.e., combine attack data with inventory, vulnerabilities, and network status data). It provides information to the CND Analyst community that facilitates the execution of selected COAs to mitigate and respond to attacks directed at the GIG. The ESG will collect, process, and store sensing environment information (raw, processed, correlated, alert, etc.) and make that information available for use to the CND UDOP.

2.6.5.4 (U) Technology Timelines

(U//FOUO) Figure 2.6-4 contains preliminary technology timelines for this IA System Enabler. These are the results of research completed to date on these technologies. These timelines are expected to evolve as the RCD and the research of technologies related to these capabilities continues.

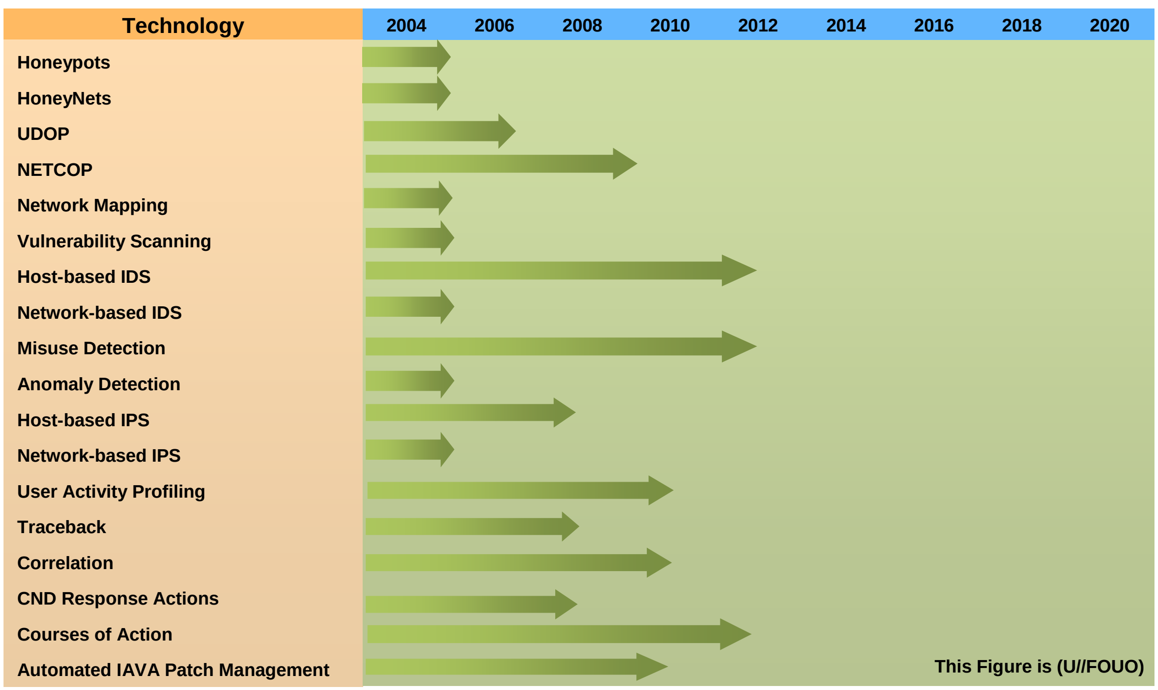


Figure 2.6-4: (U) Technology Timeline for Network Defense and Situational Awareness

2.7 (U) MANAGEMENT OF IA MECHANISMS AND ASSETS

(U//FOUO) Management of IA Mechanisms and Assets encompasses the policies, procedures, protocols, standards, and infrastructure elements required to reliably support initialization and full lifecycle management of IA mechanisms and assets. IA Mechanisms are persistent data constructs that support key IA services including identity, privilege, keys, and certificates. IA Assets are devices/software that perform an IA function. Some IA assets are:

- (U//FOUO) Cryptographic Devices (including devices providing data in transit/data-at-rest protection and protection of management and control information)
- (U//FOUO) Cross-Domain Solutions, Firewalls, Guards
- (U//FOUO) Call Trace/lawful Intercept Systems
- (U//FOUO) Intrusion Detection/Prevention Systems
- (U//FOUO) Audit Management Systems
- (U//FOUO) Virus Protection Software
- (U//FOUO) Key Generation/Management Systems
- (U//FOUO) Policy Enforcement Points (including devices that control access to information).

2.7.1 (U) GIG Benefits due to Management of IA Mechanisms and Assets

(U//FOUO) The Information Assurance constructs used to support Management of IA Mechanisms and Assets provide the following benefits to the GIG:

- (U//FOUO) Secure management of persistent IA constructs (e.g., identity, privilege, policy, key/certificate)
- (U//FOUO) Secure management of devices/software that performs an IA function
- (U//FOUO) Prevention of establishment of false identities, rogue Communities of Interest (COI)s, etc.
- (U//FOUO) Elimination of manual keying, configuration, and inventorying of IA assets
- (U//FOUO) Support for compromise recovery of IA mechanisms and assets
- (U//FOUO) Standardized protocols and common data packaging formats to address the complications of managing numerous IA-enabled enterprise entities.

2.7.2 (U) Management of IA Mechanisms and Assets: Description

(U//FOUO) Management of IA Mechanisms and Assets focuses on providing management and control of security data, processes, and resources. The security of management and control data, process and resources is the focus of the Assured Resource Allocation enabler.

(U//FOUO) The Security Management infrastructure is comprised of components, services, and products provided by external systems and within the system. Examples of products provided by a Security Management Infrastructure (SMI) include:

- (U) Unique identities for all GIG entities and COIs
- (U) Symmetric keys
- (U) Public keys
- (U) X.509 certificates
- (U) New or updated software-based cryptographic algorithms, operating systems, application software updates and patches
- (U) Virus update files.

Examples of services that must be provided by the GIG SMI include:

- (U) Identity Management
- (U) Privilege Management
- (U) Key Management
- (U) Certificate Management
- (U) Configuration Management of IA Devices and Software
- (U) Inventory Management of IA Devices
- (U) Compromise Management of IA Devices
- (U) Audit Management.

2.7.2.1 (U) Identity Management

(U//FOUO) Identity management is the capability to unambiguously associate unique assured digital identities with individuals (a.k.a., human), named groups (e.g. Organizational Domains, Operational Domains, COIs), devices, and services. Assured identities are made available to processes and functions that create, modify, or enforce policy and privileges and, therefore, must be guaranteed to represent the real GIG entity. Due to the criticality of the assured digital identity, the infrastructure that provides identity management must ensure the confidentiality, integrity, and availability of the identity registration processes, equipment, configurations, registries, and databases that it uses to operate.

(U//FOUO) The scope of identity management includes the entire lifecycle of an identity from creation, maintenance of information associated with an identity, revocation, and retiring of the identity. For named groups, identity management also includes updating the mapping of individual identities to the group. Identities must be persistent in the GIG; they cannot expire, be overwritten, or reset by events in the GIG. In fact, the identity registered for an individual is unique and remains constant despite changes of that individual's name or other attributes.

2.7.2.1.1 (U) Identity Creation

(U//FOUO) The process of creating an assured digital identity is called registration. Human registration includes the process of performing identity proofing, establishing a unique ID and initial user profile, and creating an authentication token. The authentication token may be a personal token or device management key that will later be used to authenticate that identity. At a minimum, the digital identity consists of an identifier (e.g., serial number or user name) and an associated set of attributes (for a human user, attributes may include password, PIN, public/private key pair, fingerprint, and retinal scan.) that can be used to authenticate the identity when access is requested. Assured identities must be nonforgeable to prevent masquerades.

(U//FOUO) Registration of individuals establishes and maintains a user profile that refers unambiguously to an identified entity. The identification information verified (e.g., passport, birth certificate) or collected (e.g., biometrics) during the identity proofing is maintained as identity data in the user profile. The identity proofing method used to register the individual is also maintained in the user profile and used as a factor in an access control decision. Identity proofing mechanisms for individuals could range from no proof of ID presented during registration to presenting multiple picture IDs in person. Identity proofing for devices and services will require different standards and processes than those for users.

(U//FOUO) In addition to GIG users, all managed GIG devices and services will have an assured identity. Currently devices have a serial number or a Media Access Control (MAC) address associated with them, based on their Network Interface Card (NIC). This will evolve to a nonforgeable identity in the future so that individual devices can be identified with their configurations, software, hardware, and firmware. Unique identities for managed devices will also enable the management infrastructure to more accurately keep track of GIG resources and more effectively manage devices.

(U//FOUO) Identity proofing of devices and processes will differ from that for individuals. For example, proofing of a device may require examination by a competent authority to determine whether it is a National Security Agency (NSA)-certified Type-1 device, a FIPS-level 1 device, or an uncertified device. A check of the device serial number, manufacturer's equipment number, etc., before putting the device into the GIG may also be appropriate. The result would be included in the registration profile of the device. In addition, the registration process may have to verify the pedigree of the device or service to avoid connecting potentially compromised devices or services to the GIG.

(U//FOUO) Registration requires a heterogeneous system based on open standards for identity management that focus on non-proprietary mechanisms and procedures. Methods will be required for real-time enrollment and authorization of entities in the GIG as well as archiving, binding, and auditing their identities and credentials.

2.7.2.1.2 (U//FOUO) Identity Maintenance

Information associated with an identity must be maintained as events occur that change the attributes of the entity the identity represents. For example, an individual may change their name. The user profile for the individual must then be updated to reflect the new name for the individual. Other events that may require user profiles to be updated include:

- (U//FOUO) A new authentication token is received by the individual
- (U//FOUO) An authentication token is compromised
- (U//FOUO) An individual is added to or removed from a named group.

2.7.2.1.3 (U//FOUO) Retiring of the identity

(U//FOUO) Identities could become obsolete for a variety of reason including:

- (U//FOUO) An individual no longer will be operating on the GIG
- (U//FOUO) A named group is no longer needed
- (U//FOUO) A device is destroyed.

(U//FOUO) Under any of these conditions the identity would be retired, but not deleted. Identities would be archived to allow the continued analysis of historical transactions involving that identity. As a result, the Identity Management Infrastructure must be able to archive and restore identities.

2.7.2.2 (U) Privilege Management

(U//FOUO) The GIG model is based upon massively distributed resources and services that are to be dynamically and selectively drawn from (e.g., information Pull) and utilized by a large and diverse user population. In addition, this same user population will be given the capability to influence and modify (e.g., information Post or Push) the GIG-resident databases. Due to these inherent capabilities of the GIG, a globally robust and secure way is required to manage the privileges assigned to a GIG entity. The synchronization of privileges across the GIG is essential to support collaborative sessions that do not overstep policy-mandated sharing boundaries. The potentially vast GIG user base combined with the tremendous range of sensitivity/classification of future GIG-resident resources makes the privilege management function of utmost importance.

(U//FOUO) The GIG's Privilege Management Infrastructure (PMI) needs to be an evolution of and improvement upon traditional techniques. In general, utilization of some computer-based resources or applications has always required both the authentication (verification of identity) and authorization (verification of privilege) of a potential user. Traditionally, authorization employed an Access Control List (ACL) that was held internal to and controlled by the application itself. The most recent concepts for privilege management enable the authorization/privilege verification process to be drawn outside of individual applications. This paradigm is essential for the robust and efficient operation of the future GIG.

(U//FOUO) Privilege management in the GIG must be scaleable. Privileges will be needed in a timely fashion and consistent with their valid and authorized requirements. Potential conflicts and inconsistencies between the various sources of authority will require the development of GIG-wide arbitration entities so as to arrive at universally acceptable privilege attributes before multiple users or entities enter into any joint missions. It is anticipated that many groups (e.g., COIs) will manage their own privileges.

(U//FOUO) In all cases, the base mechanism for communicating privileges needs to be consistent. However, the set of privileges granted will vary from entity to entity. As a result, the assured identities of an entity will be associated (cryptographically bound) with one or more sets of privileges, likely a separate set for each role and COI to which the entity belongs or supports. The group of bound privileges to an assured identity would be part of a User Profile.

(U//FOUO) Privilege management must support the following operational concepts and environmental conditions:

- (U//FOUO) RAdAC Model
- (U//FOUO) Multiple Security Domains
- (U//FOUO) Temporary Mission Needs
- (U//FOUO) Dynamic COIs
- (U//FOUO) Operation within GIG Network of Networks Context
- (U//FOUO) Trusted Transport/Distribution/Synchronization

- (U//FOUO) Role-based Privileges.

(U) The roles of privilege management in supporting each of these are described in the following sections.

2.7.2.2.1 (U//FOUO) Privilege Management Role in RAdAC Assured Sharing Model

(U//FOUO) One of the core concepts of the GIG, essential to enabling on-the-fly and situational-agile access-privilege control, is the RAdAC model, shown in Figure 2.7-1.

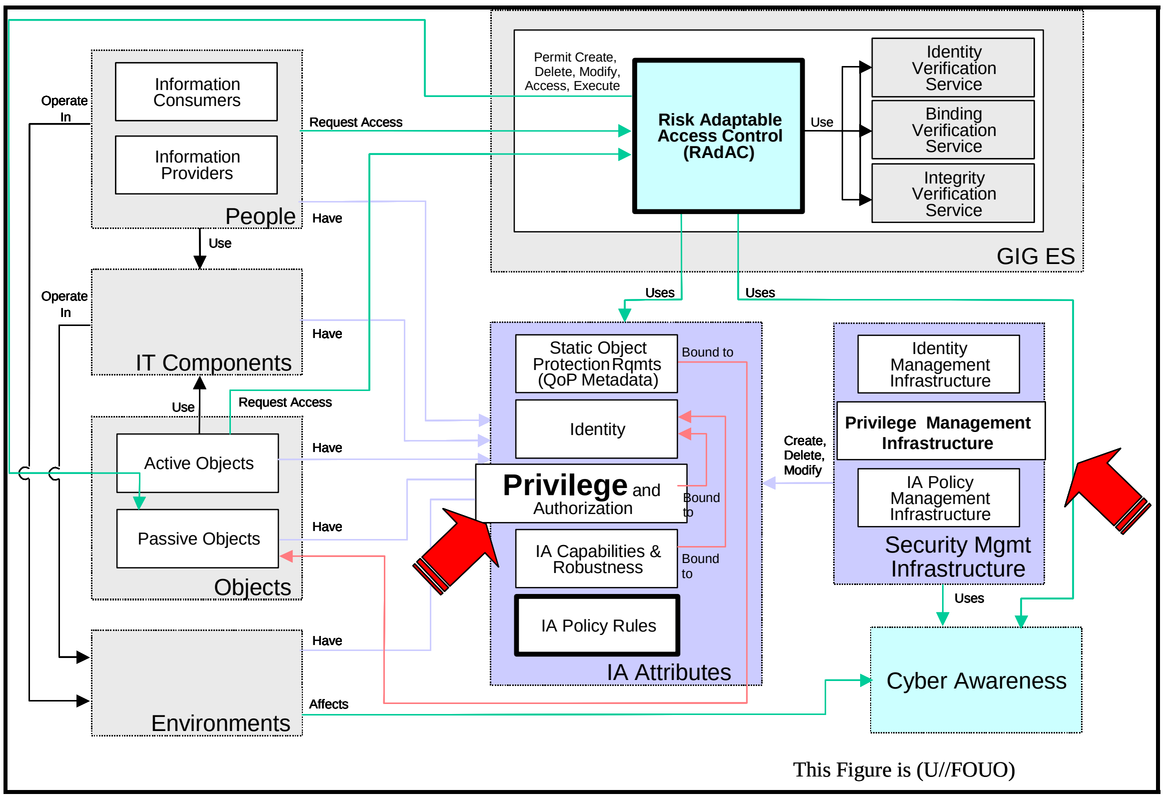


Figure 2.7-1: (U//FOUO) Assured Sharing Context Diagram Emphasizing Privileges

(U//FOUO) As shown, the Privilege IA attribute and the PMI that manages it are key elements in the notional flow of the RAdAC process. Moreover, not only do users have privilege authorization; so do the active objects they access (e.g., applications and services) and the IT components that they use (e.g., routers, servers, clients). As discussed in Section 2.2, Policy-Based Access Control, the privileges of all entities involved in a transaction are evaluated before granting access. For example, the user may have the right privileges to access information, but the client through which the user is accessing the GIG may be in a less secure environment or may not have the required set of IA capabilities or security robustness to permit access. In this case, access would be denied.

(U//FOUO) The privileges that any future GIG PMI must manage will include privileges to not only gain knowledge of distributed GIG resources, but also to act upon those resources, e.g., read, write, modify, delete, and share various information entities, be they data, software, or policy. Thus the PMI needs to be multidimensional in this sense.

2.7.2.2.2 (U//FOUO) Accommodations of Multiple Classification Levels

(U//FOUO) One of the basic features that will drive the function of a GIG PMI will be the need to accommodate multiple levels of classification. This applies both to the situation in which a single user is operating within the context of a single session on the GIG (in which case that lone user's clearance level shall dictate the classification level up to which the user may gain access) and also to the more likely scenario in which multiple users of potentially different clearance levels must collaborate in order to accomplish a joint mission. Collaboration requires joint situational awareness based on the lowest common denominator of clearance-based privileges so as to not violate or overstep any classification-limited sharing boundaries.

(U//FOUO) An example of how this might work is a Multi-Level Security (MLS) system with the following type of Mandatory Access Control scheme. Each piece of information is given a security label (as metadata), which includes classification level (e.g., unmarked, unclassified, FOUO, NATO-restricted, confidential, secret, top secret, compartmented), and each subject user has a clearance which specifies the classification level the user is permitted to access.

(U//FOUO) A potential security policy (i.e., privilege) designed to stop information leakage while maximizing sharing would permit formation of collaborative sessions among a group of users at a level equal to or lower than the lowest common set of privileges. Users with MLS devices could form multiple concurrent sessions at different levels, and they could shift between levels based on the current access policy (e.g., read down/write up). Users with single-level devices would have to either end one session to access information at a different level (as determined by RAdAC), transfer the information through a cross-domain solution (assuming the information was at an appropriate level as determined by RAdAC), or request information only at or below their current level. This would allow users to read targets with lower classifications than their own clearance and to write to targets with higher classifications. Thus, effective collaboration within a coalition of users with varying clearance levels is accommodated.

2.7.2.2.3 (U//FOUO) Adaptation to Temporary Mission Needs

(U//FOUO) Exception handling to support temporary mission needs would be supported by a policy that designates when exceptions are allowed (given human intervention) for access to GIG resources not normally available based upon an entity's current privileges. In this case, it may be necessary for the GIG privilege management infrastructure to enable temporary (time-limited) alterations to individual privileges to support the special mission. In this example, an entity would be temporarily provided the privilege to assert precedence or priority for access to certain GIG resources during a specific mission. This will require that the PMI provide for globally-available notification of this increase in privilege and that it be automatically validated system-wide.

2.7.2.2.4 (U//FOUO) Support of Dynamic COIs

(U//FOUO) Future COIs that operate within the context of the GIG are likely to be not only diverse but dynamic from day to day as single coalition partners arrive and depart from participation in collaborative sessions. This may require an adaptive and agile scheme to assign and modify individual and coalition-wide privileges to meet needs.

2.7.2.2.5 (U//FOUO) Operation within the GIG Network of Networks Context

(U//FOUO) The GIG will evolve as a collection of networks that are tied together, each with its own Network Operations Center (NOC). These networks include the Transformational Satellite (TSAT) network, the Global Information Grid – Bandwidth Expansion (GIG-BE) network, the mobile/wireless JTRS Joint Tactical Radio System (JTRS), and Net Centric Enterprise Services (NCES). These first three are the fundamental transport networks over which GIG services (such as NCES) will be accessed.

(U//FOUO) Each of the transport networks and enterprise services will have its own defined populations of users and operational entities, all of whom will require managed sets of privilege attributes. Privilege management can be thought of as occurring at three basic levels (from lowest to highest)—local administration, Service/Transport network operations and administration (as described above), and GIG-wide operations and administration.

(U//FOUO) Control of the various networks will be done through action of each relevant NOC, with an envisioned GIG-wide NOC eventually coming into being (though not entirely supplanting the intermediate NOC roles). Division of network control among these requires a commensurate PMI functionality across these networks. This mandates the tying together and cross-awareness of the various PMI level actions so that privileges are jointly adjudicated.

2.7.2.2.6 (U//FOUO) Trusted Transport/Distribution/Synchronization

(U//FOUO) In support of essentially static COIs, the GIG PMI will need to have the ability to securely transport (with integrity and confidentiality) and distribute privileges to all necessary parties before collaborative sessions can start. If a coalition membership becomes dynamic with resultant modification of joint privileges, then there will be a need for timely and synchronous distribution across the GIG of sharing privilege modifications.

2.7.2.2.7 (U//FOUO) Support of Role Based Privileges

(U//FOUO) In addition to individual-based privilege management, there will likely be the need for role-based privileges in the GIG. A role is defined by a specific set of tasks that require a set of privileges in order to be performed. Typical roles in the GIG would be IA security manager with policy-setting privileges, network administrator with NOC control privileges, and mission-specific roles.

2.7.2.3 (U) Key Management

(U//FOUO) Cryptography is one of the fundamental IA mechanisms used to protect the GIG, and cryptography cannot be implemented correctly without key management. Key management is one of the fundamental aspects of Information Assurance. The full lifecycle of key management includes:

- (U) Key Management Practice Statement
- (U) Key Ordering
- (U) Key Generation and Labeling
- (U) Key Packaging and Distribution
- (U) Storage, Backup and Recovery
- (U) Revocation and Destruction.

2.7.2.3.1 (U) Key Management Practice Statement

(U//FOUO) A Key Practice Statement is a document that describes the process of handling and controlling cryptographic keys and related material (such as initialization values) according to key policy. It details key management functions and parameters available to authorized users.

(U//FOUO) Key Management Plans are written for systems that use keys. Such plans need to be compatible with the Key Practice Statement. However since the GIG is not being built or operated as a single, consolidated system, it is not reasonable to expect that there will be a single GIG Key Management Plan. Rather, each constituent component of the GIG (e.g., GIG-BE, TSAT, JTRS, and end user systems connecting to the GIG) must have a key management plan. Component key management plans will adhere to established key management standards and approved architecture. Appropriate authorities for completeness and consistency with other component key management plans must review these plans, and any discrepancies must be resolved prior to operation. For example, if the GIG-BE key management plan makes assumptions about the duty and ability of End Cryptographic Units (ECUs) to protect keys, then no ECU should be connected to the GIG-BE unless its key management plan clearly states how it protects those keys sufficiently to meet GIG-BE assumptions.

2.7.2.3.2 (U) Key Ordering, Generation, Labeling, Packing, and Distribution

(U//FOUO) The first phase of a key's life supports the request and delivery of key material to the intended recipient. This begins by ordering of the key material by a user who is authorized to request keys. Once an order is verified to come from a valid requestor, an authorized key source can generate the key material, label the key and its attributes, package the key in a manner compatible with delivery protocol, and distribute the key to the specified recipient.

(U//FOUO) Distribution may be either physically or electronically. Electronic delivery includes the use of NSA-approved benign techniques for encrypted, over-the-network (OTNK) key distribution by a direct network connection between the keying source and the intended receiving device.

(U//FOUO) Keys and algorithms used by GIG components must be only those approved by authorized key sources (e.g., NSA). Keys can be either locally-generated or provided by a central authority. Keys provided by a central authority must be validated before being used. Locally-generated keys must be generated only through approved processes and equipment and must be used only within defined constraints.

(U//FOUO) Any cryptographic algorithms used in the GIG must be approved by authorized sources. No ECU shall use an algorithm unless it can be validated as approved by an authorized source and not be modified in an unapproved way.

2.7.2.3.3 (U) Storage, Backup and Recovery

(U//FOUO) Key storage is performed at the authorized key source and at the receiving device. Keys must be stored securely on an ECU. Even if stored in software rather than on a dedicated hardware device, the key must be stored so that it can neither be extracted easily by an attacker (including attackers' software agents), nor modified without detection in an unauthorized way.

(U//FOUO) At the trusted key source, the key must be backed up to support the following:

- (U) Decryption of stored enciphered information
- (U//FOUO) Continuity of operation when the key is not readily available due to conditions such as crypto period expiration, key corruption, or permanent departure of the key owner
- (U//FOUO) Key recovery.

(U//FOUO) The key management infrastructure must be able to identify all ECUs impacted by a key compromise and ensure the rapid recovery of operations by supporting key compromise recovery mechanisms with the affected ECUs.

2.7.2.3.4 (U) Revocation and Destruction

(U//FOUO) At times it is necessary to revoke a key before its expiration. This may occur because its use is no longer needed, or the key may have been compromised. Revocation of a key that has not been compromised does not require its destruction, but the key management infrastructure must support a mechanism for notifying GIG entities that the key can no longer to be used.

(U//FOUO) All GIG components must have a way of destroying keys when circumstances require it. When a key is destroyed, it must not be possible for an adversary with physical possession of the hardware on which the key resided to recover any parts of the key. Key destruction mechanisms must be designed in such a way as to minimize the chance of unintended or accidental destruction.

2.7.2.4 (U) Certificate Management

The following main phases define the certificate life cycle management process;

- (U) Adherence to CPS (Certificate Practice Statement)
- (U) Registration/Enrollment
- (U) Certificate Creation
- (U) Certificate Distribution
- (U) Certificate Retrieval
- (U) Certificate Expiration
- (U) Certificate Revocation.

2.7.2.4.1 (U) Adherence to CPS

(U) The Certificate Practice Statement lists the services supported and practices used throughout the Certificate Life Cycle. These services include registration, creation, distribution, storage, retrieval, revocation, and other supporting sub-services. This process is used to govern the operating principles at the various levels – which include individual components, enclaves, enterprise or the entire infrastructure (e.g., Public Key Infrastructure [PKI]). The adherence to the CPS should be auditable, and the appropriate measures should be place to account for activities related to Certificate Management phases.

2.7.2.4.2 (U) Registration

(U) Registration process starts when an end-entity requests a Registration Authority (RA) to issue a certificate. Depending on the Certificate Practice Statement, Certificate Policy, and privileges associated with the requested certificate, the identity verification may require a physical appearance or submission of appropriate authorization documentation. The same is true for registering devices, except that devices do not make appearances, but rather have a representative to act on their behalf.

(U) RAs are a critical element within the infrastructure. The assurance level attained within the infrastructure is dependent on the accuracy of their actions and their adherence to established policies. The higher the level of assurance required within the infrastructure, the more stringent the identification process. The RA provides the new User's information to the Certificate Authority (CA) which then creates a key pair and a Certificate.

(U) Clearly, the Registration Authority plays a very critical role in the overall security and integrity of the infrastructure. If RAs do not adhere to established procedures and properly verify identify or accurately enter other personal information, they put the entire infrastructure at risk.

2.7.2.4.3 (U) Certificate Creation

(U) The CA has responsibility for certificate creation, regardless of where the key is generated. A certificate binds an entity's unique distinguished name (DN) and other additional attributes that identifies an entity with a public key associated with its corresponding private key. The entity DN can be an individual, an organization or organizational unit, or a resource (web-server/site). Appropriate certificate policies govern creation and issuance of certificates. The public key needs to be transmitted securely to the CA in case it was generated elsewhere by a party other than the CA. Certificates can be used to verify a digital signature or for encryption purposes.

(U) There are several groups working on the standards for a specific application area, and hence there exist a number of certificate profiles or formats for different requirements. SPKI, PGP, and SET formats are popular versions. Most of them derive from the X.509 Version 3.0 specification. A typical X.509 Certificate contains several standard fields and additional policy-related extension fields.

(U) Though certificates enable the PKI, there are several privacy issues surrounding an individual's certificate usage [2]. Requests and subsequent distribution of keys and certificates require secure transmission modes. The IETF PKIX working group has defined management and request message format protocols (CMP/CRMF) specifically for this purpose. Alternatives such as Public Key Cryptography Standards (PKCS) also exist.

2.7.2.4.4 (U) Certificate Distribution

(U) Certificate Distribution involves securely and easily making the certificate information available to a requestor. This can be done through several techniques, including out-of-band and in-band distribution, publication, centralized repositories with controlled access, etc. Each has its own benefits and drawbacks.

(U) Depending on the client-side software, certificate usage, privacy and operational considerations, the information requirements and distribution methods vary. Several protocols are available that facilitate secure distribution of certificates and revocation information. For example, enterprise domains widely use LDAP repositories with appropriate security controls along with in-band distribution through S/MIME based e-mail. This hybrid approach maximizes the benefits. Even within the repository model several configurations like direct-access, inter-domain replication, guard mechanism, border, and shared repositories are possible and often used.

2.7.2.4.5 (U) Certificate Retrieval

(U) Certificate Retrieval involves access to certificates for general signature verification and for encryption purposes. Retrieval is necessary as part of the normal encryption process for key management between the sender and the receiver. It is also necessary for verification, as a reference where the certificate containing the public key of a signed private key is retrieved and sent along with the signature or is made available on demand.

(U) It is imperative to have an easy and simple mechanism to retrieve certificates. Otherwise the whole infrastructure will introduce unacceptable inefficiency. Validation is performed to ensure a certificate has been issued by a trusted CA in accordance with appropriate policy restrictions and to verify its integrity and validity (whether expired/revoked) before its actual use. In most cases all this is achieved transparently by the client-software before cryptographic operations using the certificate are carried out.

2.7.2.4.6 (U) Certificate Expiration

(U) Certificate Expiration occurs when the validity period of a certificate expires. Every certificate has a fixed lifetime and expiration is a normal occurrence. A certificate can be renewed provided the keys are still valid and remain uncompromised. When renewed, a new certificate is generated with a new validity period. In this case, the same public key is placed into the new certificate. Alternatively, a certificate update can also be done to create essentially a new certificate, with a new key pair and new validity period. Certificate update, like key update must take place before the certificate expires. In this case, the policy restrictions may remain the same as that of the expiring certificate.

2.7.2.4.7 (U) Certificate Revocation

(U) Certificate Revocation is the cancellation of a certificate before its natural expiration. Several situations warrant revocation. For instance, it could be due to privilege changes for the certificate owner, key loss due to hardware failure, private key compromise, etc. Cancellation per se is an easier process when compared to properly notifying and maintaining the revocation information. The delay associated with the revocation requirement and subsequent notification is called revocation delay. This is clearly defined in the Certificate Policy, because it determines how frequently or quickly the information is broadcast and used for verification.

(U) When there is a subscriber compromise, all subscribers within the entire infrastructure can be exploited until the compromise is detected. Therefore, compromises of individual subscribers must be dealt with quickly and efficiently, with new keys generated as appropriate. Concurrently, the Compromised Key List (CKL) would need to be updated. Should the CA itself be compromised, all CA subscribers would need to be rekeyed and new Certificates created.

2.7.2.5 (U) Configuration Management of IA Devices and Software

(U//FOUO) Configuration Management (CM) of IA devices and software provides the ability to manage and control the IA equipment and software components that provide the framework for the IA infrastructure or provides IA services within the GIG. Examples of these components include ECUs, trusted platforms, trusted software, and software elements that provide or support IA functionality (e.g., anti-virus updates). An ECU is a device, normally a component of a larger system, which contains cryptographic functionality, provides security services to the larger system, and from the viewpoint of a supporting management infrastructure, is the identifiable component with which a desired management transaction can be conducted. Management transactions can also be conducted with IA software elements, which include either embedded or stand-alone software functionality that supports GIG IA services.

(U//FOUO) Configuration Management activities involve the distribution, handling, and storage of software, data packages, and policy used by the IA devices or software to control dynamic mission parameters needed to establish their various operational configurations.

(U//FOUO) The types of configuration changes considered to be part of IA CM, as compared to the CM performed as part of traditional network management, include:

- (U//FOUO) Cryptographic algorithm updates
- (U//FOUO) IA device feature updates
- (U//FOUO) Virus (malware) detection/prevention updates.

(U//FOUO) Cryptographic algorithm updates are needed to support the GIG 2020 Vision in which ECUs must be able to change algorithms to meet new interoperability or mission requirements. This change—adding support for new algorithms; ceasing support for outdated algorithms; switching algorithm modes—must happen only under authorized conditions. That is, the units must have a way to recognize that an authorized entity is telling it to change algorithm support, and the unit must then be capable of acting on that request. Unauthorized attempts to change algorithm support must be rejected.

(U//FOUO) Coalition interoperability is one example in which the ability to upload different cryptographic algorithms is beneficial. Currently, coalition interoperability is generally accomplished by providing U.S. systems to partners. However, this has some negative side effects; notably, the coalition partner has direct access to U.S. hardware and software. It also requires the logistics step of physically transporting that hardware to the coalition partner's location and training coalition partners on equipment operation. In the 2020 system, the GIG must be capable of interoperating with coalition partners' existing systems. By uploading algorithms in the U.S. equipment that are compatible with the coalition partners' equipment, there would be no need to share U.S. equipment, because our equipment would interoperate with the coalition equipment. The GIG must interoperate with coalition partners, while simultaneously providing a high assurance U.S.-only capability. The ability to communicate on one channel of the equipment using the coalition partner's algorithms and on another channel with U.S. algorithms satisfies warfighter needs.

(U//FOUO) U.S. Policy sometimes requires a reduced set of features in IA enabled devices used overseas. The CM characteristic that supports device feature updates enables the capabilities of the device to be tailored to the feature set appropriate for the operating environment.

(U//FOUO) Today, the control and management of virus (malware) detection/prevention capability is currently performed locally at a virus detection server. These server activities include application update and configuration per policy, virus signature pull operations from the external source to the parent server, and configuring the update (push) and scan policy for clients connected to this parent server. The parent virus detection server can also gather statistics and scan results based on CND policy settings.

(U//FOUO) In the future, as the GIG migrates from edge-to-edge encrypted network to a converged Black Core (end-to-end) network, it will become more critical that trusted, and up to date virus detection applications be resident on GIG clients. This client application-based malware code defense will form a last critical barrier in this type of encrypted core architecture where IPv6 tunneled packets are not decrypted and checked at the traditional DMZ firewall network boundary. This type implementation will make scalability, distribution of updates, and synchronization important between the parent virus detection server and the large number of GIG client that could be affected by this type of malware attack.

(U//FOUO) CM operations are accomplished by information exchange between GIG management systems (local or remote) and target devices and software components. The following paragraphs highlight a number of the critical aspects associated with security management of the GIG's IA devices and software.

(U//FOUO) The management infrastructure is responsible for the packaging, delivery, and control of software/firmware packages/dynamic policy parameters. A software/firmware/anti-virus update package must have been developed, tested, and evaluated and validated before distribution. Distribution of validated packages could be operator initiated or automated as a result of configuration changes determined by CND operations.

(U//FOUO) The CM infrastructure will verify the signature and will assume authority for the management and distribution of the package or policy. It will be responsible for commanding and performing any required preprocessing (e.g., common data formatting). As part of distribution to the target IA devices/software, the management infrastructure signs and encrypts, as required, the configuration information.

(U//FOUO) Once the targeted IA devices/software receives a configuration package from the management system, it must validate the source of the package and verify the package's data integrity. This implies that the proper trust anchors have been installed. (Trust anchors and management authority are established as part of the initialization process.) Handling and storage of configuration information at a device also requires an ability to read and act upon version information contained in the package. Finally, the element must also provide feedback status information to its directing management system.

(U//FOUO) The CM infrastructure must monitor and maintain compliance of the IA devices/software configurations with the current security and configuration policies. If discrepancies are found, distribution of the current configuration packages would be initiated. The target IA devices/software provides version/status information in response to query traffic from its validated and authorized management system. In summary, in order to enable the GIG IA Vision, existing configuration management functionality must be enhanced in the areas of source authentication support, transfer confidentiality/integrity, and version management.

2.7.2.6 (U) Inventory Management

(U//FOUO) Inventory Management provides the ability to exchange machine identification, status, version, and network topology information between the target IA devices and the management infrastructure. During the manufacturing or initialization process, GIG devices will be given a Unique Identifier that conforms to an Identity Management Standard. When queried by an authorized management system, the device will output its identification information and configuration information. The device configuration information being queried often is sensitive. Therefore, confidentiality of the inventory information must be maintained in this process. In addition, queries and requests will need to be authenticated by the device before processing.

(U//FOUO) The Inventory Management infrastructure uses the status information to support higher level accounting, tracking and network location system as well as providing information and data support to network visualization tools, cyber situational awareness, and Computer Network Defense (CND) systems. Use of this information is described in Section 2.6.

2.7.2.7 (U) Compromise Management of IA Devices

(U//FOUO) The GIG infrastructure must support Compromise Management of IA Enabled Equipment. IA Enabled Equipment is considered compromised when its integrity or confidentiality is no longer assured. This might occur through such mechanisms as exploitation of a vulnerability by a worm, or through physical loss of equipment possession. Compromise Management includes:

- (U//FOUO) Detection – The determination, through audit or network sensors, that a compromise may have occurred. This is described in Section 2.6.
- (U//FOUO) Investigation – Confirmation of the status of IA Enabled Equipment. This involves communicating with the equipment and confirming its configuration and state.
- (U//FOUO) Isolation – The active steps taken to ensure that the compromised equipment is isolated from the rest of the GIG that compromised keys are invalidated, and that the equipment is cleared of all sensitive information and rendered benign.
- (U//FOUO) Restoration - The initialization, reconfiguration, and reconnection to the GIG of equipment which is suspect, due either to loss of control or known compromise.

2.7.2.8 (U) Audit Management

(U//FOUO) The GIG infrastructure must also support audit management. Audit management processes include:

- (U//FOUO) Ability to configure IA audit data gathering per policy
- (U//FOUO) Local collection of auditable events that identify the source of the audit data
- (U//FOUO) Secure storage and transfer of the logged information from the device to the management infrastructure
- (U//FOUO) Ability to analyze the audit data to identify significant events.

(U//FOUO) All these process must be supported with integrity services to assure accuracy of the audit data. Audit data provides a means to detect any events that resulted in a security breach of the GIG system. Once audit data is gathered from IA components and correlated, the audit management infrastructure provides Security Operations/Administrations personnel the following:

- (U//FOUO) A means of independent review and examination of records to determine the adequacy of system controls to ensure compliance with established policies and operational procedures
- (U//FOUO) Information needed to alter the use of resources to improve system performance
- (U//FOUO) A source of data that can be used to identify an individual, process, or event associated with any security-violating event.

(U//FOUO) In summary, to enable the GIG IA Vision, existing audit management functionality must be enhanced by incorporating unique identifiers for authenticated individuals/devices into audit event records, resource utilization recording, trusted time tagging, secure storage, transfer integrity, and risk-based access to audit records.

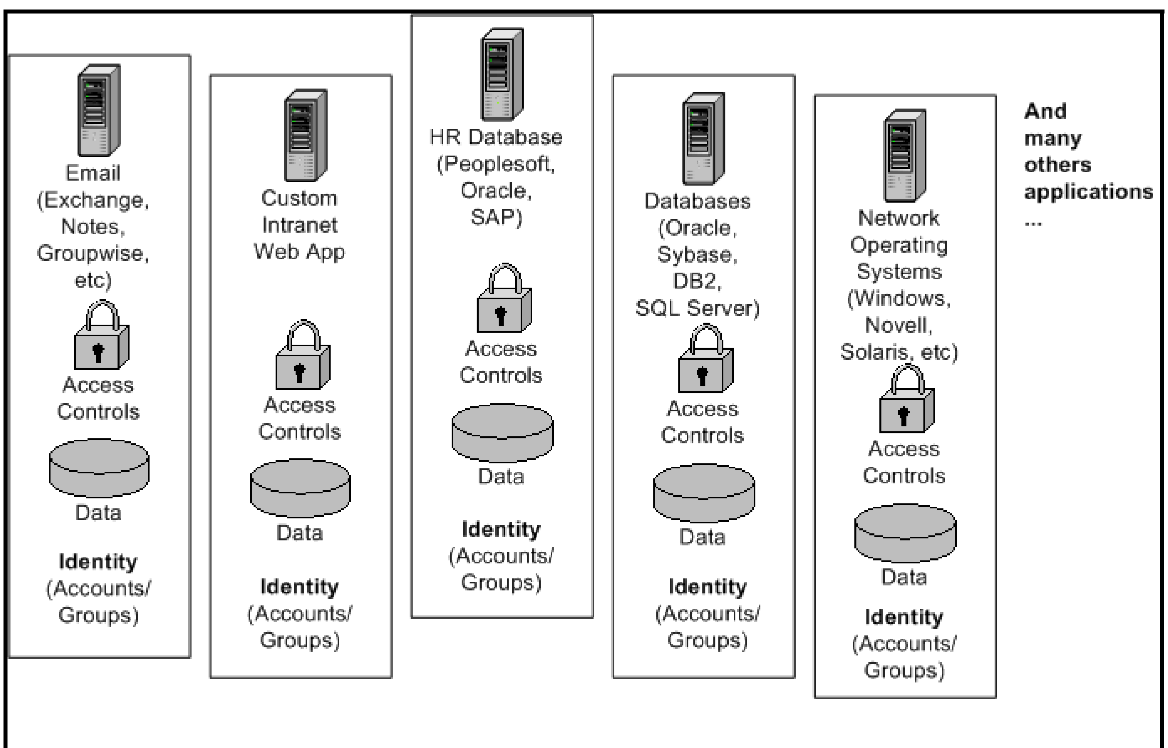
2.7.3 (U) Management of IA Mechanisms & Assets: Technologies

2.7.3.1 (U) Identity Management

2.7.3.1.1 (U) Technical Detail

(U) Identity management is essentially the process of creating and maintaining entity accounts and credentials. Throughout an enterprise, an entity may have many identities; a user account on a UNIX system, another account on the mail server, digital certificates for Secure Socket layer (SSL) access, and a smart card for building access. Each identity is used for access control decisions on each independent system. For example a UNIX account name will not mean anything to the SSL-enabled web server.

(U) Historically not only are the identities not related to each other, they are managed independently as well. Independent management can lead to many problems for users and enterprises. An entity will have to be enrolled and provisioned in each system to which it requires access, a time consuming activity. Further, when an entity's access to the GIG has been revoked (i.e., a person quits their job, a device is destroyed, etc.), each system needs to terminate the entity's account. Account termination may be a very low priority activity, causing inactive accounts to exist for some time after the user has left the enterprise. An example of the multiple identities a user might have is shown in Figure 2.7-2.



This Figure is (U)

Figure 2.7-2: (U) Example of Multiple Identities Assigned to a Single User

(U) Identity Management provides a means to unify disparate identity data stores. By controlling identity information in a single location, user accounts need to be created only once. Users also have an easier time administering their own profiles and other account information as they only have one location to make modifications. Also, non-human entities that require identity information can be controlled in a more consistent and automated fashion. Further, access control (i.e., privilege management, Section 2.7.2.2) information can be bound to the identity, whether stored locally or remotely.

(U) The scope of identity management includes the entire life cycle of an identity from creation, maintenance of information associated with an identity, revocation, and retiring of the identity. For named groups, identity management also includes updating the mapping of individual identities to the group. Identities must be persistent in the GIG and not expire, or be overwritten or reset by events in the GIG. In fact, the identity registered for an individual is unique and remains constant despite changes of that individual's name or other attributes.

(U) Identity Management systems typically only handle identities within an enterprise. However, there may be times (such as when dealing with different programs) when identity information needs to be exchanged outside of the native enterprise boundaries. Federated management is the concept of a user being allowed to use the same identity across multiple enterprise identity management systems. For instance, a warfighter could sign into an external resource with the same identity information and credentials as he/she would normally use for their native resources.

2.7.3.1.2 (U) Usage Considerations

(U) Seamless integration of identity management comes at a cost. The enterprises that form a federation must trust each other. Effectively, one partner must trust the other in order to vouch for the validity of a given user. This type of trust may be a bit much for programs to bear in the initial years of integrated Identity Management use. As time goes on and Identity Management practices and standards evolve, there will likely be greater trust in the technologies and programs allowing Federated Identity Management to take hold.

(U) Federated identity management is one of the biggest concerns when implementing identity management in the GIG. In order to make identity management grow to something larger than an enclave-level service, federations will need to be formed between programs, services, and agencies. Unfortunately, it is unclear where and how federations should be created. Should coalition partners be part of the GIG federation? Will multiple federations exist? How will these federations interoperate? These are important questions that will need to be answered as GIG programs integrate identity management.

(U) The DoD has invested much money in the Common Access Card (CAC) system. CAC cards are a smartcard based systems for providing a unique identity for any entity within the DoD. The CAC platform provides most of the DoD with a common identity system that can be leveraged for GIG-wide identity systems.

(U//FOUO) While not a perfect solution, the CAC system is a good first start. Presently CAC cards are not fully used as electronic identity tokens. They are still generally used as physical badges, since many of the back-office systems that could take advantage of CAC cards are still being developed. However, a program, started in summer 2004, called Federated Identity Cross - credentialing System/Defense Cross-credentialing Identification System (FiXs/DCIS) is attempting a large leap forward. FiX/DCIS, is a pilot program designed to use CAC tokens to test federated access between DoD programs and DoD contractors. The program, co-sponsored by the Defense Manpower Data Center and the Office of the Secretary of Defense, will provide practical insight into using CAC cards in an Identity Management system. More info can be found at http://www.fegc.org/.

(U) The standards for identity management are still emerging. Identity management promises to be an important technology in the next decade. As such, many big industry and government organizations have stepped up to assist in standards development. However, as with any high-profile standards process, some vendors disagree on the technical details and end up creating separate and competing standards. This will continue to be a problem until the industry matures further.

(U) SAML – Security Assertion Markup Language

(U) From the SAML Technical Overview on oasis-open.org

(U) “The Security Assertion Markup Language (SAML) standard defines a framework for exchanging security information between online business partners.

(U) More precisely, SAML defines a common XML framework for exchanging security assertions between entities. As stated in the SSTC charter, the purpose of the Technical Committee is:

(U) …to define, enhance, and maintain a standard XML-based framework for creating and exchanging authentication and authorization information.

(U) SAML is different from other security systems due to its approach of expressing assertions about a subject that other applications within a network can trust. What does this mean? To understand the answer, you need to know the following two concepts used within SAML.

(U) Asserting party

(U) The system, or administrative domain, that asserts information about a subject. For instance, the asserting party asserts that this user has been authenticated and has given associated attributes. For example: This user is John Doe, he has an email address of john.doe@acompany.com, and he was authenticated into this system using a password mechanism. In SAML, asserting parties are also known as SAML authorities.

(U) Relying party

(U) The system, or administrative domain, that relies on information supplied to it by the asserting party. It is up to the relying party as to whether it trusts the assertions provided to it. SAML defines a number of mechanisms that enable the relying party to trust the assertions provided to it. It should be noted that although a relying party can trust the assertions provided to it, local access policy defines whether the subject may access local resources. Therefore, although the relying party trusts that I'm John Doe – it doesn't mean I'm given carte blanche access to all resources."

(U) Available from http://www.oasis-open.org/

(U) SPML – Service Provisioning Markup Language

(U) SPML is intended to facilitate the creation, modification, activation, suspension, and deletion of data on managed Provision Service Targets (PSTs). It is the only real standard of import that deals explicitly with the act of provisioning. Provisioning is a core component of Identity Management, but unfortunately most of the standards work has been in the direction of privilege management.

(U) Available from http://www.oasis-open.org/

(U) XACML – eXtensible Access Control Markup Language

(U) From http://www.oasis-open.org/committees/download.php/2713/Brief_Introduction_to_XACML.html

(U) *"XACML is an OASIS standard that describes both a policy language and an access control decision request/response language (both written in XML). The policy language is used to describe general access control requirements, and has standard extension points for defining new functions, data types, combining logic, etc. The request/response language lets you form a query to ask whether or not a given action should be allowed, and interpret the result. The response always includes an answer about whether the request should be allowed using one of four values: Permit, Deny, Indeterminate (an error occurred or some required value was missing, so a decision cannot be made) or Not Applicable (the request can't be answered by this service)."*

(U) Available from http://www.oasis-open.org/

(U) Liberty Alliance

(U) The Liberty Alliance is an industry-created standards setting body. Project Liberty is largely concerned with Federated Identity Management. Their standards include ID-FF (the Identity Federation Framework), ID-WSF (Identity Web Service Framework), and ID-SIS (a collection of Identity Services Interface Specifications).

(U) Available from http://www.projectliberty.org/

2.7.3.1.2.1 (U) Implementation Issues

(U//FOUO) Creation of GIG-wide Identity Management Schema – When implementing an identity management system, a schema describing users, their properties, and profiles must be created. This schema can vary significantly from enterprise to enterprise. For the GIG, a schema should be developed that encompasses DoD-wide needs. Further, systems need to be designed to handle potential future schema modifications. Whatever identity management schema that is developed in the near term will likely need revision after a few years of deployed use.

(U) Evolving Standards – The standards revolving around identity management are still evolving. While the standards settings bodies are (generally) attempting to maintain backward compatibility, it is still critical to design systems that can adapt to the changes. From a software engineering standpoint, this underscores the need to write modular code that abstracts the user from the underlying standards. However, with the use of web services (a direction for most identity management systems) modularity is already a core construct, so changing standards should have less impact.

(U//FOUO) Integration of Privilege Management – Identity management and privilege management go hand in hand (sometimes they are even referred to as the same concept). Due to the complexity of the GIG, these concepts get separate treatment since they could be managed at different levels. For example, GIG-wide identities may require a centrally controlled construct. However, privileges may be managed at a local level to support COIs. In any case, in order to be fully functional, an Identity Management system must integrate seamlessly with the Privilege Management system.

(U//FOUO) Supporting Directory Infrastructure – An Identity Management system will need to have a supporting directory structure to store the identity information. Many programs already have existing installed directories, whether it be LDAP, Active Directory, NDS, or some other system. GIG-compliant programs may chose to either implement a new directory system from scratch or leverage existing infrastructures. Any directory system, however, must comply with the concept of least privilege for identity information stored in the directory store. That is, unlike the general concept of an open directory, Identity Management directories will contain information that is sensitive or classified in nature and must be protected as any other data store would be.

(U//FOUO) Delegated and Dynamic Management – For an Identity Management system as large as will be required for the GIG, delegated management is an important yet difficult requirement. Depending on the situation (wartime vs. peacetime), location (in the Pentagon vs. in the field), and other factors, the scope and speed of changes to the identity management system by an actor may vary significantly. The identity management must reflect the chain of command in a service, allowing those in control of a warfighter or GIG entity to make changes (add privilege, add profiles, etc) to the identity information of that entity. Further, when there are large state changes (such as going to war), the Identity Management system will have to automatically and securely update privileges of entities to wartime privileges.

2.7.3.1.2.2 (U) Advantages

(U) Identity Management provides two major advantages to an enterprise. The first advantage is cost savings. Rather than create a user account in many systems, a user can be enrolled in a central identity management system. This cuts down the man-time it takes to get a user up and running in an enterprise. Further, the self-service aspect of an identity management system can allow users to manage their own profiles and credentials. This can reduce help desk calls for things like lost passwords and name changes.

(U) The second advantage is security. By managing all users and entities through a common mechanism, policies can be applied uniformly across all actors. Accounts can be terminated in a timely manner. Access can be granted from a central location, enabling auditable policy enforcement. In general, Identity Management can get rid of the mish-mash of accounts and roles in an organization.

2.7.3.1.2.3 (U) Risks/Threats/Attacks

(U//FOUO) As with any system that tries to unify data storage and allow for distributed access, the movement to provide Identity Management in the GIG creates a new risk for the GIG.

(U//FOUO) Identity Theft – By unifying identity information, even at a enclave level, identity management system can become a central location for hijacking an entity's identity. This is commonly called identity theft. However, in the context of the GIG, the ramifications are more severe for the enterprise than to an individual. An attacker who could subvert the identity management system could take on the identity of a trusted entity. This would allow the attacker to operate with the privileges of the subverted account.

(U//FOUO) Denial of Service – Another concern as identity management becomes more pervasive is denial of service attacks. Many identity management architectures rely on a central host(s) to be available to either a) validate the identity, b) obtain a list of attributes or c) check to see if the identity token has been revoked. If the central hosts can be disabled, the identity management may cease to work. Generally speaking, this will cause problems throughout the system that is relying on the identity management system. Disabling the identity management system may affect a large number of systems. This makes the identity management infrastructure a weak point in the enterprise and should be protected as such.

2.7.3.1.3 (U) Maturity

(U) Currently, Identity Management is assessed a maturity level of Early (TRLs 1 - 3). While standards exist and there have been limited programs adopting the technology, it is not ready for deployment. The vast majority of what is dubbed Identity Management is actually privilege management. Privilege management is a more robust technology with many more vendors providing solutions today. However, strict identity management is immature.

2.7.3.1.4 (U) Standards

(U) The standards for identity management (Table 2.7-1) are still emerging. It promises to be an important technology in the next decade. As such, many big industry and government organizations have stepped up to assist in standards development. However, as with any high-profile standards process, some vendors disagree on the technical details and end up creating separate and competing standards. This will continue to be a problem until the industry matures further.

Table 2.7-1 (U) Identity Management Standards

This Table is (U)	
Name	**Description**
OASIS Standards	
SAML Core	E. Maler et al. *Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML)*. OASIS, September 2003. Document ID oasis-sstc-saml-core-1.1. http://www.oasis-open.org/committees/security/.
SAML Gloss	E. Maler et al. *Glossary for the OASIS Security Assertion Markup Language (SAML)*. OASIS, September 2003. Document ID oasis-sstc-saml-glossary-1.1.http://www.oasis-open.org/committees/security/.
SAMLSec	E. Maler et al. Security Considerations for the OASIS *Security Assertion Markup Language (SAML)*, OASIS, September 2003, Document ID oasis-sstc-saml-sec-consider-1.1. http://www.oasis-open.org/committees/security/
SAMLReqs	Darren Platt et al., SAML Requirements and Use Cases, OASIS, April 2002, http://www.oasis-open.org/committees/security/.
SAMLBind	E. Maler et al. *Bindings and Profiles for the OASIS Security Assertion Markup Language (SAML)*. OASIS, September 2003. Document ID oasis-sstc-saml-bindings-1.1. http://www.oasis-open.org/committees/security/.
SPML – Service Provisioning Markup Language	SPML is intended to facilitate the creation, modification, activation, suspension, and deletion of data on managed Provision Service Targets (PSTs). It is the only real standard of import that deals explicitly with the act of provisioning. Provisioning is a core component of Identity Management, but unfortunately most of the standards work has been in the direction of privilege management. http://www.oasis-open.org/committees/documents.php
SPML-Bind	OASIS Provisioning Services Technical Committee., SPML V1.0 Protocol Bindings, http://www.oasis-open.org/apps/org/workgroup/provision/download.php/1816/draft-pstc-bindings-03.doc, OASIS PS-
XACML – eXtensible Access Control Markup Language	From http://www.oasis-open.org/committees/download.php/2713/Brief_Introduction_to_XACML.html
Liberty Alliance	
ID-FF	Identity Federation Framework Available from http://www.projectliberty.org/
ID WSF	Identity Web Service Framework Available from http://www.projectliberty.org/
ID SIS	Identity Services Interface Specifications Available from http://www.projectliberty.org/
This Table is (U)	

2.7.3.1.5 (U) Cost/Limitations

(U) Deployment of Identity Management systems can have high up-front costs. The initial planning on how identities will be standardized within an enterprise will require coordination from any party that will be affected by the transition. Applications need to be either retooled to use the identity management system, or middleware needs to be used to interface legacy systems with the Identity Management infrastructure.

2.7.3.1.6 (U) Dependencies

(U) In a vacuum, identity management is only marginally useful. It really becomes useful when coupled with Privilege Management to provide authorization information to applications. Any Identity Management system will rely on a Privilege Management system to provide real value.

2.7.3.1.7 (U) Alternatives

(U) The alternative to Identity Management is the status quo with respect to user and entity information. Some programs may choose to not integrate into a common identity management framework. While these programs may be able to avoid integration in the near-term based on cost and security considerations, eventually user requirements will drive the need for full integration as identity management matures. Users will expect to have a single interface for all account management. Further, GIG-wide CND mechanisms will be integrated into GIG Identity Management systems to enable centralized attack sensing and defense. Systems not leveraging the GIG identity management system may lose these protections.

2.7.3.1.8 (U) Complementary Techniques

(U) Managing identity is only part of the entire access control domain. Privilege must also be managed in order to complete the picture. It is not enough to simply prove who an entity is; a system must be able to provide authorization for that entity to perform actions. Managing this authorization is the core of privilege management and integral with any systems Identity Management architecture.

2.7.3.1.9 (U) References

(U) [Reed] The Definitive Guide to Identity Management; Reed, Archie; Realtimepublishers.com; 2002-2004.

2.7.3.2 (U) Privilege Management

2.7.3.2.1 (U) Technical Detail

(U) The goal of privilege management is to allow fluid access to legitimate resources. Privilege management technologies had not evolved significantly until only recently. That is because collaborative networks and computing environments have grown very large and complex within the past decade, consequently requiring improved solutions towards identification, authentication, and authorization solutions.

(U) In the early computing environment days and even up to the present day, privileges were implemented via ACLs that were commonly found on the major operating systems. They were used to control access to files, directories, and services on either local hosts or mainframes. As the computing environments evolved, there was a progression towards utilizing scripts to automate rules towards managing a user's privileges to resources. More recently, during the past decade in particular, we have seen an even more sophisticated drive towards concepts of Role-based Access Control (RBAC) and even infrastructure-based concepts such as PMI. These significant concepts as well as the standards and technologies that surround them are described below.

2.7.3.2.1.1 (U) Rules-Based Authorization Schemes

(U) Rules are provided as run-time processes that dynamically determine outcome based on privileges. Rules can include complex Boolean operations, using an interpretive language or scripting language to define rules.

(U) Instead of aggregating all permissions within predefined roles (roles are detailed in the next sub-section), some enterprises have chosen to take advantage of rule-based processing capabilities of provisioning systems and WAM (Web Access Management) products. Rule-processing engines examine and evaluate user attributes and privileges, and make outcome decisions on the fly. This functionality permits more dynamic actions to be taken during processing instead of relying on the ability to map out every possibility in advance. Rule-based processing may be more dynamic than roles, but at the same time requires that business processes be accurately understood.

2.7.3.2.1.2 (U) Roles-Based Authorization Schemes

(U) More recently, research has focused on RBAC. In the basic RBAC model, a number of roles are defined. These roles typically represent organizational roles such as secretary, manager, employee, etc. In the authorization policy, each role is given a set of permissions, i.e., the ability to perform certain actions on certain targets. Each user is then assigned to one or more roles. When accessing a target, a user presents his role(s), and the target reads the policy to see if this role is allowed to perform the action.

(U) There are several fairly new standards to choose from; however, there are minimal implementations or compatibility with the standards to really make them useful. The National Institute of Standards and Technology (NIST) offers an RBAC reference model and The Organization for the Advancement of Structured Information Standards (OASIS) offers eXtensible Access Control Markup Language (XACML)—an XML specification for expressing policies for information access over the Internet.

(U) The NIST core RBAC offers a good overview of what is desired in an RBAC solution at http://csrc.nist.gov/publications/nistbul/csl95-12.txt.

(U) The NIST component defines five basic data elements:

- (U) Users – An entity that uses the system
- (U) Roles – A job function within the context of an organization
- (U) Permissions – Approval to perform an operation on one or more objects
- (U) Objects – Can be many things; for example, an entry in a target system (such as an account), a network resource (a printer), an application (a procurement), a policy (password policies), and so on
- (U) Operations – Various and unbounded but including customer-defined workflow processes such as a password reset, the addition, modification, or removal (deletion) of user accounts, and specific data about those accounts; importantly, it should be possible to delegate these operations to other users

(U) Hierarchical RBAC

(U) Hierarchical RBAC requires the support of role hierarchies, whereby senior roles acquire the permissions of their juniors, and junior roles acquire the user membership of their seniors.

(U) The NIST standard recognizes two types of role hierarchies.

- (U) General Hierarchical RBAC—Arbitrary orders and relationships between roles serve as the role hierarchy.
- (U) Limited Hierarchical RBAC—Restrictions are placed on the role hierarchy. Typically hierarchies are limited to simple structures such as trees or inverted trees.

(U) Although General Hierarchical RBAC introduces potential problems of hierarchy loop detection and prevention, it is seen as the most useful. In an RBAC solution, consider that occupants of the same roles at different locations in an organization will need access to different underlying systems. This allows the same role (say, development engineers) to be given access to different systems based on differing values in the role occupant's profile. So while all development engineers need access to source control, it is likely that those in one office or working on one product may need access to a different source control system from those in another office or working on a different project. To solve this problem, a parameterized permission object can be used. A single permission Source Control Access might be used. However the mappings from that object into the connected systems (that is, source control systems) would vary based on a user's location attribute or on the project attribute.

2.7.3.2.1.3 (U) Privilege Management Infrastructure (PMI)

- (U) PMI is the information security infrastructure that assigns privilege attribute information, such as privilege, capability, and role, to users and issues. One options is to manages manage privileges by using the X.509 Attribute Certificate (AC). The function of the PMI is to specify the policy for the attribute certificate issuance and management. Then, the PMI carries out the AC-related management functions, such as issuing, updating, and revoking an attribute certificate based on a specified policy.

(U) Although Attribute Certificates were first defined in X.509(97), it was not until the fourth edition of X.509 (ISO 9594-8:2001) that a full PMI for the use of attribute certificates was defined. A PMI enables privileges to be allocated, delegated, revoked, and withdrawn electronically. A PMI is to authorization what a PKI is to authentication. Table 2.7-2 summarizes these relationships.

Table 2.7-2: (U) Comparisons of PKI and PMI

This Table is (U)

Concept	PKI Entity	PMI Entity
Certificate	Public Key Certificate (PKC)	Attribute Certificate (AC)
Certificate Issuer	Certificate Authority (CA)	Attribute Authority (AA)
Certificate User	Subject	Holder
Certificate Binding	Subject's Name to Public Key	Holder's Name to Privilege Attribute(s)
Revocation	Certificate Revocation List (CRL)	Attribute Certificate Revocation List (ACRL)
Root of Trust	Root Certification Authority or Trust Anchor	Source of Authority (SOA)
Subordinate Authority	Subordinate Certification Authority	Attribute Authority (AA)

This Table is (U)

(U) A public key certificate (PKC) is used for authentication and maintains a strong binding between a user's name and his public key, while an attribute certificate (AC) is used for authorization and maintains a strong binding between a user's name and one or more privilege attributes. The entity that digitally signs a public key certificate is called a CA, while the entity that digitally signs an attribute certificate is called an Attribute Authority (AA). The root of trust of a PKI is sometimes called the root CA while the root of trust of the PMI is called the Source of Authority (SOA). CAs may have subordinate CAs that they trust, and to which they delegate powers of authentication and certification. Similarly, SOAs may delegate their powers of authorization to subordinate AAs. If a user needs to have his signing key revoked, a CA will issue a Certificate Revocation List (CRL). Similarly if a user needs to have his authorization permissions revoked, an AA will issue an attribute certificate revocation list (ACRL).

(U) PMI systems need to provide the following functionalities:

- (U) Assigning attributes to a user
- (U) Creating an X.509 attribute certificate
- (U) Issuing, updating, revoking, searching, and publishing attribute certificate
- (U) Validating an attribute certificate and making an access control decision
- (U) Supports ID/Password or PKC authentication method
- (U) Applying RBAC model to access control framework
- (U) Supports push/pull model in an attribute certificate usage
- (U) Supports flexible system architecture by providing independent DMS.

2.7.3.2.2 (U) Usage Considerations

2.7.3.2.2.1 (U) Implementation Issues

(U) There are various implementations between Authority Management, Policy Management, and other components within the PMI.

(U) X.509 supports simple RBAC by defining role specification attribute certificates that hold the permissions granted to each role, and role assignment attribute certificates that assign various roles to the users. In the former case, the AC holder is the role, and the privilege attributes are permissions granted to the role. In the latter case the AC holder is the user, and the privilege attributes are the roles assigned to the user.

(U) Another extension to basic RBAC is constrained RBAC. This allows various constraints to be applied to the role and permission assignments. One common constraint is that certain roles are declared to be mutually exclusive, meaning that the same person cannot simultaneously hold more than one role from the mutually exclusive set. Another constraint might be placed on the number of roles a person can hold or on the number of people who can hold a particular role.

(U) X.509 only has a limited number of ways of supporting constrained RBAC. Time constraints can be placed on the validity period of a role assignment attribute certificate. Constraints can be placed on the targets at which a permission can be used and on the policies under which an attribute certificate can confer privileges. Constraints can also be placed on the delegation of roles. However many of the constraints (e.g., the mutual exclusivity of roles) have to be enforced by mechanisms outside the attribute certificate construct (i.e., within the privilege management policy enforcement function).

2.7.3.2.2.2 (U) Advantages

(U)The challenges of role-based access control will continue to be the contention between strong security and easier administration. For stronger security, it is better for each role to be more granular—thus to have multiple roles per user. For easier administration, it is better to have fewer roles to manage.

(U) The creation of rules and security policies is also a complex process. Depending on the situation within the enterprise, there will be a need to strike an appropriate balance between the two.

(U) PMI-based solutions have the advantage of existing or emerging infrastructures such as PKI. But on the other hand, management schemes for PMI and the attribute certificates are considered to be complex and more challenging compared to the other privilege management schemes.

2.7.3.2.2.3 (U) Risks/Threats/Attacks

(U//FOUO) PMI will need to provide complete and accurate audit of authorization activity. It will be necessary to have both attributable receipts of the transaction, and of the contributing activities, such as the authorization decision.

(U//FOUO) XML is a good prospect for creating these receipts, digitally signing them, and submitting them to a notarization service to enhance their non-repudiation ability. Vendors are currently providing capabilities for creating, storing, and managing non-repudiated records via XML encryption and digital signature.

2.7.3.2.3 (U) Maturity

(U) Privileges, and specifically Attribute Certificates, have some basis as an X.509 extended standard. However, the management of rules and roles-based access specifications are still left to proprietary implementations. There are enabling technologies such as SAML and XACML to assist in the development of RBAC as well as rules-based applications, and some COTS vendors have in fact implemented solutions based on these underlying technologies. Given the availability of prototypes that prove working concepts, this technology is assessed as Emerging (TRLs 4 - 6).

2.7.3.2.4 (U) Standards

The Privilege Management Standards are listed in Table 2.7-3.

Table 2.7-3: (U) Privilege Management Standards

This Table is (U)	
Name	**Description**
IETF Standards	
RFC3281	S. Farrell, R. Housley, "An Internet Attribute Certificate Profile for Authorization", IETF RFC, April 2002
ISO Standards	
ISO/IEC 9594-8	ITU-T Rec. X.509 (2000) | ISO/IEC 9594-8 The Directory: Authentication Framework
This Table is (U)	

2.7.3.2.5 (U) Dependencies

2.7.3.2.5.1 (U) Relationship of Authorization to Identity

(U) Authorization policies are rules for determining which subjects are allowed to access resources. In some cases, privacy considerations may require that some form of anonymous or pseudonymous access be supported. In most cases, however, users must first be identified in order to receive authorization to access resources.

(U) An identity system is therefore critical to establishing users' identities as the basis for authorizing access to resources. The identity infrastructure binds a unique name or identifier to a user. It also maintains a set of attributes (often in a general-purpose directory service) that supports the authentication and authorization processes. These attributes could include not only credentials, such as hashed passwords or X.509 certificates, but also information about the user, which could be referenced in an access rule.

2.7.3.2.5.2 (U) Standards Development

(U) Three classes of standards specifications can improve the interoperability of policy-based management systems:

- (U) Schemas: provide standardization in the way groups, roles, rules, and resources are described in directories and other repositories
- (U) Protocols: enable interoperability of policy decision requests and policy distribution across products from multiple vendors
- (U) Languages: provide means of codifying policy rules.

(U) The lack of policy standards has been a significant barrier to building interoperable authorization systems, but now a number of standards are being developed. In fact, several standards have recently emerged—such as SAML, XACML, Web Services Policy (WS-Policy), and XrML—creating the risk that standards will overlap.

2.7.3.2.5.3 (U) SAML – Enabling Technology

(U) SAML is intended to provide a session-based security solution for authentication and authorization across disparate systems and organizations through the use of XML SAML, which was also described earlier with Identity Management. SAML, in addition to authentication assertion, also provides Authorization Assertion, which implies the system can assert that a subject is authorized to access the object. The SAML specification also enables protocols to send and receive messages, as well as specify bindings that define how SAML message exchanges are mapped to SOAP exchanges. SAML can use multiple protocols, including HTTP, Simple Mail Transfer Protocol (SMTP), File Transfer Protocol (FTP), and SOAP.

2.7.3.2.5.4 (U) XACML – Enabling Technology

(U) SAML enables PEP-to-PDP or PDP-to-PDP communication of requests and responses for authentication, attributes, and authorization. However, SAML does not define detailed semantics for the data it carries. Roles in SAML, for example, are only text strings. OASIS left it to XACML to provide the details for attribute information or authorization information. XACML fulfills SAML's needs by providing richer semantic constructs for authorization information. Among other things, it enables use of common LDAP attributes in XML-based security protocols. But XACML does much more than this.

2.7.3.2.6 (U) Complementary Technologies

(U) Policy rules form the basis for authorization. As an extension of policy-based security, management, and networking within a single organization, standards groups (such as, the Distributed Management Task Force [DMTF], the TeleManagement Forum [TMF], the Open Group, OASIS, and a group of vendors standardizing Web services) have all been working on the definition of standardized policy objects that can be used to create or negotiate agreements, make decisions, or carry out obligations.

2.7.3.3 (U) Key Management

2.7.3.3.1 (U) Technical Detail

2.7.3.3.1.1 (U) Evolution of Key-based Equipment Technology

(U) The following are supported ECUs and their associated technologies that have evolved over the past few decades. This can be seen in Figure 2.7-3.

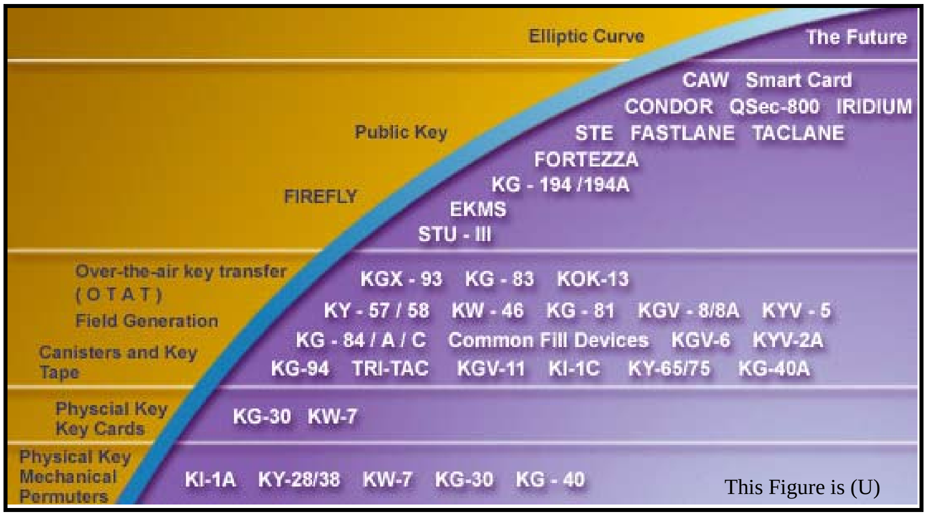


Figure 2.7-3: (U//FOUO) ECU and Technology Evolution

(U//FOUO) There is a growing need for the DoD and government enterprises to reexamine existing approaches that provision cryptographic key products and services for military, intelligence, governments, allied, contracting and business customers. It is no longer feasible or cost effective to design, develop, and field unique, independent key and certificate management systems to support the various classes of cryptographic products, as seen in Figure 2.7-4.

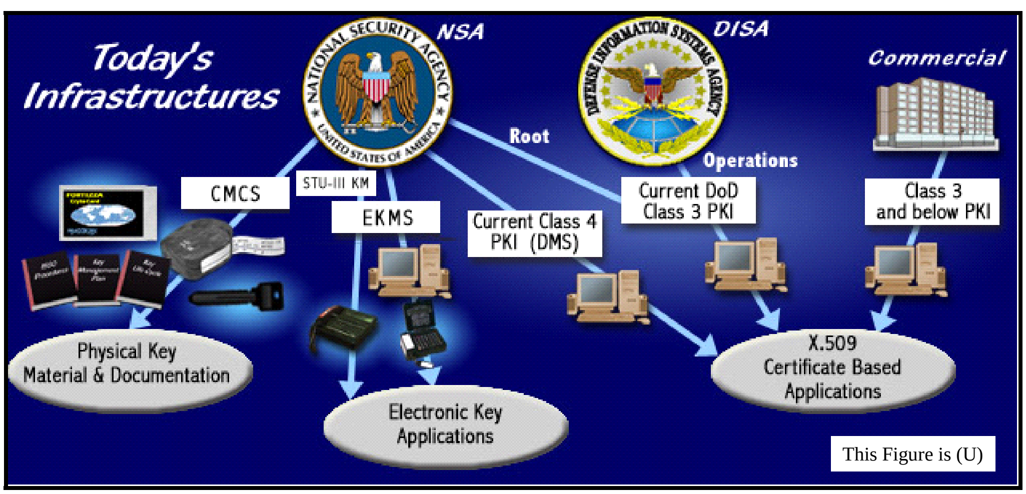


Figure 2.7-4: (U) Current Key Management Infrastructures

(U//FOUO) Advances in information technology are giving the customer a wider array of communication choices, while simultaneously necessitating wide ranges of IA solutions that are equal to the task. The current key management environment is made up of separate and independent infrastructures that provide and manage their own set of security products. These systems will become increasing cumbersome and costly as new technology and their attendant security solutions continue to advance and the resources needed to operate them decline. This key management environment, shown in Figure 2.7-4, is comprised of several unique solutions built for specific product lines. While the solutions satisfy unique security needs, they each require different tools and training in order to obtain their respective products and services, imposing an unwarranted strain on resources.

(U//FOUO) Adding a new key management capability has frequently meant creating a new, independent system to support it. The most recent example is in the public key certificate arena where independent infrastructures are being deployed to meet the demand created by the use of PKI-based security products. Continuing this approach will increasingly tax resources throughout the community.

(U) Several of the systems in Figure 2.7-4 have been in existence for a number of years and are in need of upgrade to take advantage of recent advances in communication technology. This technology area has advanced significantly in recent years, providing the market place with many new and worthwhile, applicable techniques that would greatly improve efficiency and performance.

(U//FOUO) Although created independently, the existing systems contain many common threads (e.g., registration, ordering, and distribution) that could logically be combined and offered as a unified set of processes. Not only has the key management community recognized this fact, so has the DoD Joint Staff. They have identified a unified Key Management Infrastructure (KMI) as a critical infrastructure needed to support key and certificate management approaches for mission critical, logistic, and administrative systems.

(U//FOUO) Given the critical importance of key management and the state of the current key management systems, the focus should be on developing a singular approach, using sound IA principles and modern technology.

2.7.3.3.1.2 (U) Vision of the KMI

(U//FOUO) Consequently, the NSA has launched the KMI Strategic and Architectural Planning initiative, supported by Service, Joint Staff, and contractor personnel The KMI initiative will focus on unifying the disparate key management systems within a single, modern architecture—one that is modular, flexible, and extensible. Unification will eliminate redundant resources associated with operation, maintenance, and training that will result in substantial cost savings.

(U//FOUO) The KMI will be the primary means to support the many current and future cryptographic products and services needed to conduct secure electronic transactions. Security services such as identification and authentication, access control, integrity, non-repudiation, and confidentiality become increasingly critical as the government transitions to an electronic environment. The KMI provides a means for the secure creation, distribution and management of the cryptographic products that enable these services for a wide variety of missions as seen in Figure 2.7-5.

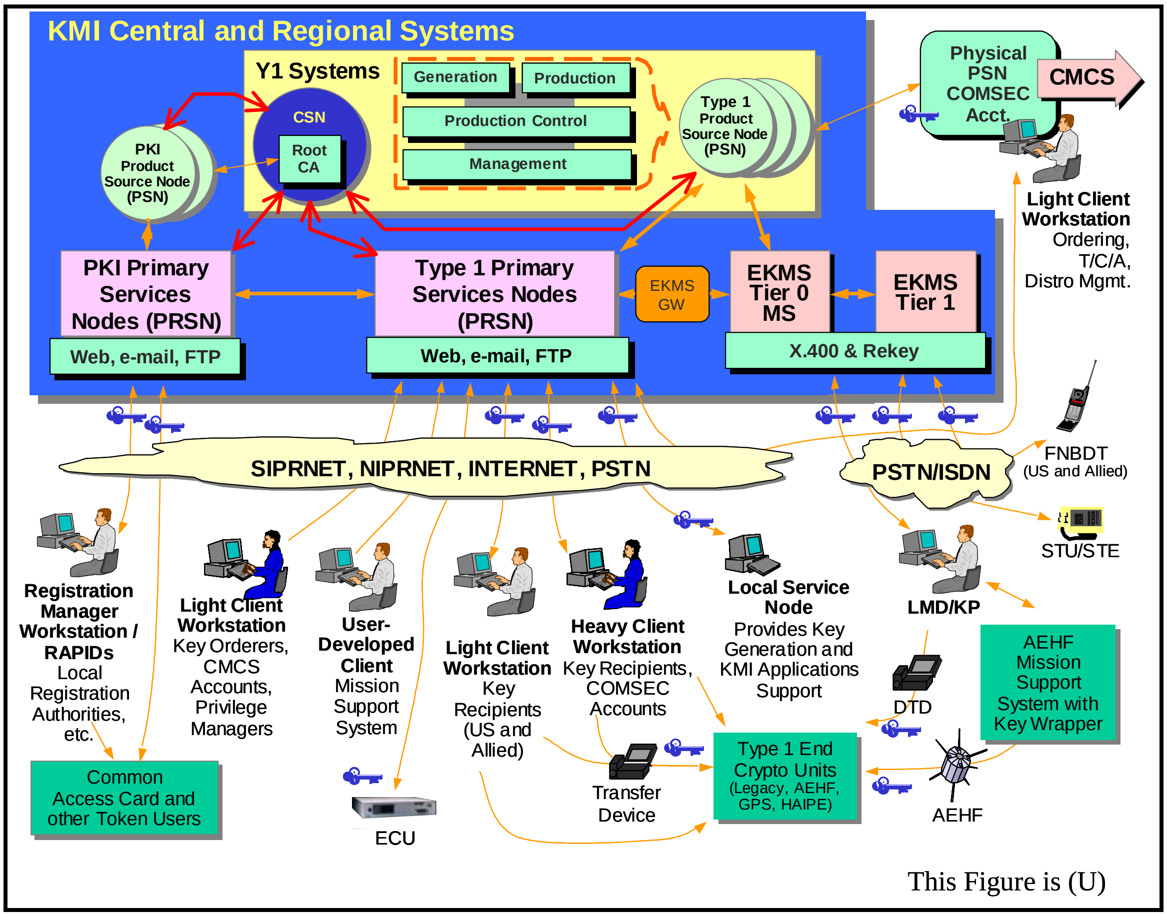


Figure 2.7-5: (U//FOUO) KMI – Envisioned Infrastructure

2.7.3.3.1.3 (U) Scope of KMI

(U//FOUO) The current key management systems service a wide variety of Departments, Services, Agencies, and Organizations within the U.S. Government and those of its allies. The common characteristic of these customers is their need to protect classified or mission critical SBU information or to inter-operate with U.S. components in doing so. As the KMI initiative evolves, its architecture will be designed, at a minimum, to continue the support of these customers' traditional requirements, as well as their growing information assurance needs for the less sensitive, but important, unclassified operational information. This means providing everything from Type 1 netted key or Class 5 certificates for classified applications to commercial Class 3 or 2 certificates for lesser needs.

(U//FOUO) Support different infrastructures such as:

- (U//FOUO) COMSEC Material Control System (CMCS)
- (U//FOUO) Electronic Key Management System (EKMS)
- (U) PKI
- (U) Government - Class 4
- (U) Government - Class 3
- (U) Commercial
- (U//FOUO) STU-III Infrastructure.

(U//FOUO) It is anticipated that requirements for support of classified applications will continue to grow as new Type 1 solutions, such as secure wireless and Global Positioning System modernization, are implemented. It is the intent of the KMI to enhance the DoD's capability to support these mission-critical requirements.

(U//FOUO) It is projected that there will be a significant increase within the DoD in the use of cryptographic applications for the conduct of unclassified and sensitive but unclassified (SBU) business transactions. Many of these applications will be obtained from commercial sources, with their keying and management services being supplied by the evolving DoD PKI or commercial service providers.

(U//FOUO) Today, the DoD is fielding two independent PKI systems supporting different assurance levels. The MISSI, or High Assurance PKI (HAPKI), supports security applications that handle medium to high value information in any environment, and the DoD. The Medium Assurance PKI (MAPKI) supports security applications that handle medium value information in a low to medium-risk environment.

2.7.3.3.1.4 (U) Key Components of a KMI

2.7.3.3.1.4.1 (U) Central Oversight Authority

(U//FOUO) The central oversight authority is the entity that provides overall key and data synchronization, as well as system security oversight for an organization or set of organizations. The central oversight authority: 1) coordinates protection policy and practices (procedures) documentation, 2) might function as a holder of data provided by service agents, and 3) serves as the source for common and system level information required by service agents (e.g., keying material and registration information, directory data, system policy specifications, and system wide key compromise and certificate revocation information). As required by survivability or continuity of operations policies, central oversight facilities may be replicated at an appropriate remote site to function as a system back up.

2.7.3.3.1.4.2 (U//FOUO) Key Processing Facilities

(U//FOUO) Key processing services typically include the following services:

- (U) Acquisition or generation of public key certificates (where applicable)
- (U//FOUO) Initial generation and distribution of keying material
- (U) Maintenance of a database that maps user entities to an organization's certificate/key structure
- (U//FOUO) Maintenance and distribution of nodal CKLs and/or CRLs
- (U) Generation of audit requests and the processing of audit responses as necessary for the prevention of undetected compromises.

(U//FOUO) An organization may use more than one key processing facility to provide these services (e.g., for purposes of inter-organizational interoperation). Key processing facilities can be added to meet new requirements or deleted when no longer needed and may support both public key and symmetric key establishment techniques.

(U) Where public key cryptography is employed, the organization operating the key processing facility will generally perform most PKI registration authority, repository, and archive functions. The organization also performs at least some PKI certification authority functions. Actual X.509 public key certificates may be obtained from a government source (certification authorities generating identification, attribute, or encryption certificates) or a commercial external certification authority (usually a commercial infrastructure/CA that supplies/sells X.509 certificates). Commercial external certification authority certificates should be cross-certified by a government root CA.

2.7.3.3.1.4.3 (U) Service Agents

(U//FOUO) Service agents support organizations' KMIs as single points of access for other KMI nodes. All transactions initiated by client nodes are either processed by a service agent or forwarded to other nodes for processing. Service agents:

- (U//FOUO) Direct service requests from client nodes to key processing facilities, and when services are required from multiple processing facilities, coordinate services among the processing facilities to which they are connected
- (U//FOUO) Are employed by users to order keying material and services, retrieve keying material and services, and manage cryptographic material and public key certificates
- (U//FOUO) Might provide cryptographic material and certificates by using specific key processing facilities for key and certificate generation
- (U//FOUO) Might provide registration, directory, and support for data recovery services (i.e. key recovery), as well as provide access to relevant documentation, such as policy statements and infrastructure devices
- (U//FOUO) Might process requests for keying material (e.g., user identification credentials), and assign and manage KMI user roles and privileges
- (U//FOUO) Might also provide interactive help desk services as required

(U//FOUO) A service agent who supports a major organizational unit or geographic region may either access a central or inter-organizational key processing facility or use local, dedicated processing facilities—as required—to support survivability, performance, or availability, requirements (e.g., a commercial external Certificate Authority).

2.7.3.3.1.4.4 (U) Client Nodes

(U//FOUO) Client nodes are interfaces for managers, devices, and applications to access KMI functions, including the requesting of certificates and other keying material. They may include cryptographic modules, software, and procedures necessary to provide user access to the KMI. Client nodes:

- (U//FOUO) Interact with service agents to obtain cryptographic key services
- (U//FOUO) Provide interfaces to end user entities (e.g., encryption devices) for the distribution of keying material, for the generation of requests for keying material, for the receipt and forwarding (as appropriate) of CKLs and CRLs for the receipt of audit requests, and for the delivery of audit responses
- (U//FOUO) Typically initiate requests for keying material in order to synchronize new or existing user entities with the current key structure, and receive encrypted keying material for distribution to end-user cryptographic devices (in which the content—the unencrypted keying material—is not usually accessible to human users or user- node interface processes).
- (U//FOUO) Can be a FIPS 140-2 compliant workstation executing KMI security software or a FIPS 140-2 compliant special purpose device.

(U//FOUO) Actual interactions between a client node and a service agent depend on whether the client node is a device, a manager, or a functional security application.

(U) Protection in KM Layers

(U//FOUO) Key Management (KM) layers that correspond to the Open Systems Interconnection (OSI) model, require assurance while exchanging data on the network between client systems and the server. One model with the KMI initiative, seen in Figure 2.7-6, depicts using several standardized protocols for security such as IPsec, TLS, and HTTP-S.

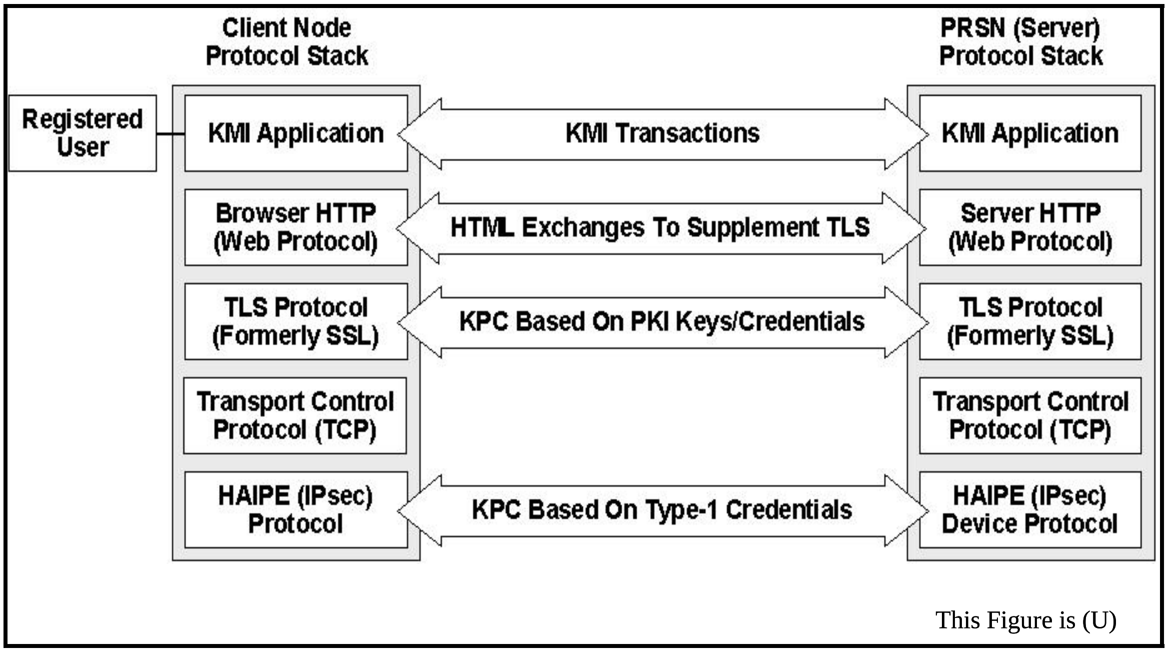


Figure 2.7-6: (U//FOUO) KMI Protected Channel Layers

2.7.3.3.1.5 (U) XML Key Management Services

(U) Online key registration, issuance, distribution, validation, and revocation services are a core feature of any network trust environment.

(U) Under the XKMS initiative (draft specification available at http://www.w3.org/TR/xkms/), the PKI industry is defining a set of XML-based services, protocols, and formats for distributing and registering public keys to support various cryptographic services—including authentication, authorization, digital signatures, content encryption, and session encryption. Principally defined by VeriSign, Microsoft, and webMethods, XKMS has already been endorsed by leading PKI providers, including VeriSign, Baltimore, Entrust, and RSA. The specification was submitted on March 30, 2001, as a Technical Note to the W3C, which has not yet created a standards-track working group to develop the specification (although creation of a formal W3C working group is likely by the end of 2004).

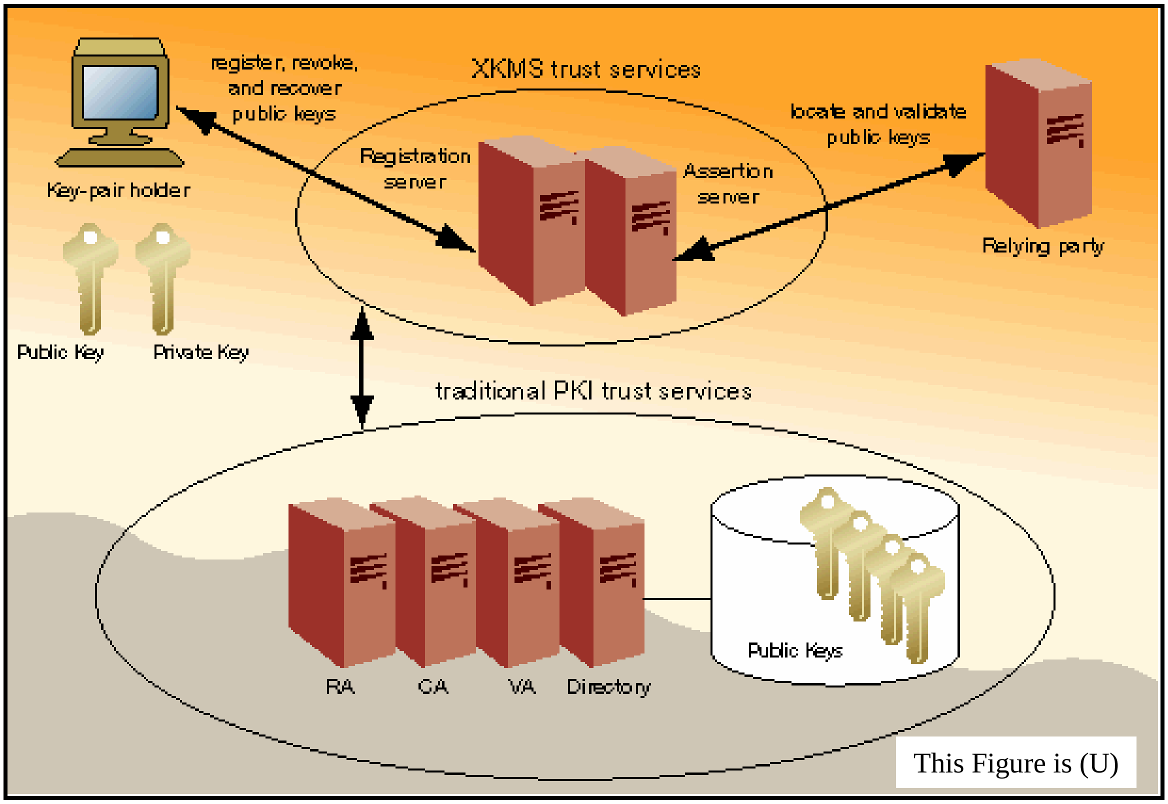


Figure 2.7-7: (U) XKMS Environment

(U) The XKMS framework consists of two services: the XML Key Registration Service Specification (X-KRSS) and the XML Key Information Service Specification (X-KISS). Registration Servers are at the heart of X-KRSS, while Assertion Servers are the hub of the X-KISS environment. Figure 2.7-7 shows a high-level functional topology of an XKMS environment that supports both the X-KRSS and X-KISS services.

(U) XKMS defines a SOAP/XML-messaging-based alternative to traditional PKI, though in many ways XKMS is designed to complement, rather than replace, established PKI standards. At the client level, XKMS defines mechanisms under which applications delegate the retrieval, parsing, and validation of X.509 digital certificates to trusted servers, thereby streamlining the configuration of client-side, trust-service business logic. XKMS requires retrofitting of today's clients and applications to support—at a minimum—such standards as SOAP, XML-DSig, XML Schemas, XML Namespaces, WSDL, and XML Encryption.

(U) The Registration Servers and Assertion Servers support all traditional PKI functions, but do so through the exchange of standardized, digitally signed XML-based messages with PKI-enabled clients. XKMS servers and clients digitally sign every message they exchange with each other via formats and mechanisms defined under XML-DSig. XKMS clients are set up to explicitly trust specific Registration and Assertion Servers, and will accept trust assertions (such as messages containing registered public keys) only if they contain valid digital signatures from those trusted servers.

(U) When deployed into a traditional X.509 PKI environment, the XKMS-enabled servers would integrate with traditional infrastructure services—including Registration Authorities, Certificate Authorities, and Validation Authorities—through established PKI X.509 (PKIX) protocols. However, the XKMS framework does not specify the need to interoperate with any external PKI. It can in fact interoperate with PKIX, PGP, or Simple PKI (SPKI) environments for such services as registering, issuing, validating, and revoking digital certificates.

2.7.3.3.1.6 (U) Constructive Key Management (CKM)

(U) CKM technology is a standards-based (X9.69 and X9.73) and patented cryptographic key management technology that resolves critical information security and information management. As more information is being created, transmitted, and stored in digital format, there is a higher percentage of information that needs to be secured. Further, the need has never been greater to identify authorized users, protect and control sensitive information assets, and restrict access to information in compliance with privacy statutes and regulations.

(U) CKM is also an authorization management system that provides logical access to individual objects. This access is enforced through encryption in a manner that efficiently supports a variety of applications, such as:

- (U) Dynamic, Assured Information Sharing
- (U) Collaboration among Communities of Interest
- (U) Digital Rights Management
- (U) Critical Infrastructure Protection
- (U) Liability Mitigation through Assured Enforcement
- (U) Data Separation
- (U) Defined Access Control to Information by Content

(U) CKM provides Cryptographically Enforced Management of Keys, Objects, and Access. CKM's Object Level Access Control (OLAC) techniques allow users to control anything that can be named, from a character, page, image, or sound in a document to a field in a database. In addition, CKM's RBAC techniques cryptographically enforce who should be able to see which piece of data or information. The approach of differentially encrypting data based on the need-to-know or need-to-share principle allows secure communication among groups of individuals with a variety of roles. Those individuals who have a legitimate need to view information have access to it, while others do not.

(U) When encrypting with CKM, users label information with Credentials that define the rights required to access the information. Users holding matching Credentials will be able to decrypt the information while those who do not will be unable to view the information. For example, a document may be labeled Proprietary or Sensitive, and it may be labeled to require certain other Credentials. Users’ Credentials are stored on Smart Tokens, which can be soft tokens or hard tokens (such as smart cards or key fobs).

(U) Behind the scenes, each Credential is associated with a public and private key pair. The public key provides encryption (writing) capabilities. The private key provides decryption (reading) capabilities. When encrypting, each of these assigned Credentials (public key values) is combined with other values and random information to construct a key. This key is used with any number of cryptographic algorithms to encrypt the information and is then destroyed. The same key will never be used again to encrypt other information.

(U) Once encrypted, the information is unreadable until it is decrypted using the same set of Credentials (private key values) and the same algorithm. Since CKM immediately destroys the key, it must later reconstruct it to decrypt the information. It does this by using a header that it attaches to the encrypted information, along with other cryptographic data retrieved from the user's Member Profile. In the header, CKM includes identifiers to the Credentials applied, but not the actual values. When decrypting, CKM attempts to retrieve the values needed to build the key from the receiver's set of Credentials. If the receiver holds the appropriate Credentials, CKM will be able to construct the key needed to decrypt the information. If not, the information will remain unreadable. This process is transparent and requires no instructions or intervention from the user.

(U) CKM technology cryptographically binds different access elements together. These elements can uniquely represent users (identity components), application processes, information, media, business rules, and scope. When these various elements are uniquely combined and mathematically proven through cryptography, the goals of content-based, role-based access and distributed information security can be achieved.

2.7.3.3.1.7 (U) IKE and ISAKMP

(U) Internet Key Exchange (IKE) is the key management protocol used with IPsec—automating the process of negotiating keys, changing keys, and determining when to change keys. IKE implements a security protocol called Internet Security Association and Key Management Protocol (ISAKMP), which uses a two-Phase process for establishing an IPsec tunnel. During Phase 1, two gateways establish a secure, authenticated channel for communication. Phase 2 involves an exchange of keys to determine how to encrypt data between the two entities.

(U) Details on IKE can be found in IETF (RFC 2409).

2.7.3.3.1.8 (U) HSM (Hardware Security Module)

(U) An HSM is a physically secure, tamper-resistant security server that provides cryptographic functions to secure transactions in applications. Acting as a peripheral to a host computer, the HSM provides the cryptographic facilities needed to implement a wide range of data security tasks. HSMs perform cryptographic operations, protected by hardware. These operations may include:

- (U) Random number generation
- (U) Key generation (asymmetric and symmetric)
- (U) Asymmetric private key storage while providing protection (security) from attack (i.e., no unencrypted private keys in software or memory)

- (U) Private keys used for signing and decryption
- (U) Private keys used in PKI for storing Root Keys
- (U) Stored value card issuing and processing
- (U) Chip card issuing and processing
- (U) Message authentication
- (U) PIN encryption and verification.

(U) HSMs offer a higher level of security than software. They are normally evaluated by third parties, such as "National Institute of Standards and Technology" (NIST), or through the Federal Information Processing Standards Publication (FIPS PUB 140-2). This level of security is required by some highly secured web applications, PKIs, and CAs.

2.7.3.3.2 (U) Usage Considerations

2.7.3.3.2.1 (U) Implementation Issues

2.7.3.3.2.1.1 (U) Key Management Policy (KMP)

(U) The KMP is a high-level statement of organizational key management policies that includes authorization and protection objectives, and constraints that apply to the generation, distribution, accounting, storage, use, and destruction of cryptographic keying material. The policy document—or documents that comprise the KMP—will include high-level key management structure and responsibilities, governing standards and guidelines, organizational dependencies and other relationships, and security objectives. [Note that in a purely PKI environment, the KMP is usually a stand-alone document known as a Certificate Policy (CP).] The scope of a KMP may be limited to the operation of a single PKI CA and its supporting components or to a symmetric point-to-point or single key center environment. Alternatively, the scope of a KMP may be the operations of a hierarchical PKI, bridged PKI, or multiple center symmetric key environments.

(U) The KMP is used for a number of different purposes. The KMP is used to guide the development of KMPSs for each PKI CA or symmetric key management group that operates under its provisions. CAs from other organizations' PKIs may review the KMP before cross-certification, and managers of symmetric key KM infrastructures may review the KMP before joining new or existing multiple center groups. Auditors and accreditors will use the KMP as the basis for their reviews of PKI CA and symmetric key KMI operations. Application owners that are considering a PKI certificate source should review a KMP/CP to determine whether its certificates are appropriate for their applications.

2.7.3.3.2.1.2 (U//FOUO) Key Packaging

(U//FOUO) In the past, different key packaging structures and delivery protocols were developed for interaction between elements of different hierarchical tiers in the KMI. This proved to be an inflexible approach in that it constrained the spectrum of possible interactions by requiring specialized interface functionality for each communicating entity. The goal is to now provide a single packaging scheme that supports interactions between entities regardless of their placement in the key management hierarchy

(U//FOUO) The existing secure key and data packaging techniques evaluated and analyzed are:

- (U//FOUO) EKMS BET (Bulk Encrypted Transaction)
- (U//FOUO) KMS Benign Techniques Transactions
- (U) S/MIME format.

(U//FOUO) The Key Packaging design must support implementation of the security mechanisms required by key transport/delivery.

2.7.3.3.2.1.3 (U//FOUO) Key Delivery

(U//FOUO) Key Delivery is a separate and distinct method from Key Packaging. The Key Delivery method addresses situations where keys (i.e., variables in the form they are to be used in a cryptographic algorithm) are moved from one entity to another. Entities may be ECUs, elements of the KMI, or Mission Support and Management Systems (MS&MSs).

(U//FOUO) The purpose of the delivery method is to provide a common key transport standard regardless of whether the key is being distributed from a PSN to a PRSN, a DTD to an ECU, a PRSN to a workstation, etc. The method is used to provide an initial key or set of keys to an ECU or to replace keys already in use to sustain a security service. It is an application layer KM method that is independent of the underlying communications protocols and is totally self supportive with respect to protecting the key.

(U//FOUO) The Key Delivery method requires certain security services. It is structured to incorporate the best of EKMS and industry standard security mechanisms. The sender and receiver in a key delivery interaction expect the following security properties to be maintained:

- (U//FOUO) Confidentiality - The key value must not be released in transit between authorized entities, that is, the key value is only known to authorized entities.
- (U//FOUO) Source Authentication - The key is received from an authorized and verifiable source.
- (U//FOUO) Integrity - The key value is accurate, that is, the received and generated keys are identical.

(U//FOUO) Key Distribution Over-the-Air Distribution (OTAD) encompasses two processes:

- (U//FOUO) Over-the-Air Rekey (OTAR) – a cryptographic equipment takes unencrypted key from a fill device, encrypts that key, and sends it to a receiving cryptographic equipment for use in that equipment.

- (U//FOUO) Over-the-Air Transfer (OTAT) – a cryptographic equipment takes unencrypted key from a fill device, encrypts that key, and sends it to a receiving cryptographic equipment that transfers that key to a fill device. The key is then loaded into a cryptographic equipment that is not on the same network.

(U//FOUO) COMSEC equipment, such as the KYX-15, is an example of a Type-1 key distribution system. The KYX-15 is the Net Control Device for a key being used within the communications net. It enables the operator to generate a key and electronically send it to any member of the net. Since the KYX-15 is the Net Controller, it has a copy of all the keys being used in the net. Each KY-57 has a unique Key Encryption Key (KEK)[12] and at least one Traffic Encryption Key (TEK).To do an Over-the-Air Rekey (OTAR), the Net Controller generates and electronically sends a new TEK (TEK 2) encrypted in the individual user's unique KEK. All of this is managed by the KYX-15 over the net communications. This process is also known as in-band rekeying.

(U//FOUO) A key that is transferred by OTAT is also sometimes available in an unencrypted form before and after distribution. Even electronic key that is distributed via EKMS is sometimes available in an unencrypted form when loading most cryptographic equipment.

(U//FOUO) Benign Techniques are used for distributing and loading key material into cryptographic equipment that do not allow exposure of the material to any entity other than the equipment which will be consuming the material. EKMS uses benign keying techniques to support all of its own internal functions. Examples of benign technique being used today can be seen with Benign Fill FIREFLY Keys, which are used by End Cryptographic Units, Local Management Devices/Key Processor, and the Central Facility to implement benign fill. The fill is the actual process by which operational keys are to be generated, distributed, and loaded into compatible cryptographic end equipment—without human exposure. This includes the loading of all cryptographic key material into the end equipment.

(U//FOUO) EKMS is currently being modified to support a broad range of benign keying techniques while interacting with and supporting new equipment. Over time, older equipment will be replaced with newer equipment using these techniques. This is being planned and coordinated under the NSA Crypto Modernization Plan.

[12] (U) A Key Encryption Key is a key that is used to encrypt or decrypt another key that is to be transmitted or stored.

2.7.3.3.2.1.4 (U)

(U) A Federal Information Processing Standard (FIPS) or NIST Recommendation will be developed to define the acceptable key establishment schemes. The standard or recommendation will select Diffie-Hellman (D-H) and MQV key agreement schemes from ANSI X9.42, RSA key agreement and key transport schemes from ANSI X9.44, and Elliptic Curve key agreement and key transport schemes from ANSI X9.63. All three ANSI documents are currently in a draft form, but are expected to be adopted by ANSI in the near future. NIST intends to select a subset of the schemes specified in the draft ANSI standards. The scheme definition document will also include a specification for a key wrapping technique, whereby a symmetric key is encrypted using another symmetric key (e.g., an AES key is encrypted by an AES key).

2.7.3.3.2.2 (U) Advantages

2.7.3.3.2.3 (U) Risks/Threats/Attacks

(U//FOUO) Risks in key management occur from the moment a cryptographic key is generated. Manual processes that handle distribution pose the single biggest threat. In fact, studies have shown that HUMINT (Human Intelligence) is the greatest threat factor. Key storage and facilities are also areas where keys can be compromised.

2.7.3.3.3 (U) Maturity

(U//FOUO) Some of the technologies in Key Management, such as generation, initial key load and rekeying are quite mature, and have been adopted under various classified (EKMS) and unclassified (PKI) infrastructures. However, the management and distribution of crypto-material still remains in some cases a very manually intensive process. Technologies such as high assurance OTNK and similar key distribution methods are only just emerging. Many of the issues that surround technological issues of high assurance with key management practices are being addressed by the KMI initiative.

(U//FOUO) Another gap area is the lack of standards for unified key labeling, packaging, and distribution formats. The only area where some semblance of standards exist here are in the PKI (public, asymmetric keys). But none exists beyond PKI. Moreover, PKI has its own limitations with keys, such as re-keying, since PKI is certificate driven and not so much key-driven. In the Type-1 Classified arena, the key packaging and distribution processes are mainly manual processes. While they follow individual and situational-based policy, there are no standards to unify these to eliminate or reduce manual error-prone and human access vulnerabilities towards threats. Standards and technologies should include the incorporation of MLS systems and data stores to close these gaps.

(U) Based on the above, the overall maturity of Key Management is assessed overall as Emerging (TRL 4-6). Prototypes and implementations for generation and sometimes-automated distribution exist, as seen in EKMS. But standardization at the enterprise-wide level, such as fulfilling GIG-wide requirements, is yet to be developed and adopted. These, as well as infrastructure issues are key concerns that the KMI effort needs to address.

2.7.3.3.4 (U) Standards

Table 2.7-4: (U) Key Management Standards

This table is (U//FOUO)	
Name	**Description**
IETF Standards	
S/MIME	Ramsdell, B., “S/MIME Version 3 Message Specification”, RFC 2633, June 1999
MIME	Freed, N., Borenstein, N., “Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies”, RFC 2045, November 1996
CMS	Housley, R., “Cryptographic Message Syntax”, RFC 3369, June 1999
ANSI Standards	
X9.69	Framework for Key Management Extensions. This standard defines specific key management methods for controlling and handling keys
X9.73	Cryptographic Message Syntax (CMS) The Constructive Key Management technique (CKM), described in ANS X9.69, is used to encrypt objects. It may be used with CMS to encrypt a message (as the object) to a set of users sharing a common set of values (known as key components)
X9.42	Key Agreement of Symmetric Keys using Discrete Logarithm Cryptography
X9.44	Key Establishment Using Factoring-Based Public Key Cryptography
NIST Standards	
FIPS PUB 140-2 ANNEX D	Security Requirements for Cryptographic Modules Annex D: Approved Key Establishment Techniques Annex D provides a list of the FIPS Approved key establishment techniques applicable to FIPS PUB 140-2.
XKMS	XML Key Management Specification (XKMS) http://csrc.nist.gov/cryptval/140-2.htm
FIPS 171	Symmetric Key Establishment Techniques National Institute of Standards and Technology, Key Management using ANSI X9.17, Federal Information Processing Standards Publication 171, April 27, 1992 http://csrc.nist.gov/publications/fips/fips171/fips171.txt
NSA Standards	
EKMS 208	EKMS Key Distribution Functional Standard. National Security Agency, Director, National Security Agency, Ft. George G. Meade, MD. 20755-6734.
EKMS 215	EKMS Communications Requirements Standard. National Security Agency, Director, National Security Agency, Ft. George G. Meade, MD. 20755-6734.
EKMS 301	EKMS Types Dictionary Standard. National Security Agency, Director, National Security Agency, Ft. George G. Meade, MD. 20755-6734.
EKMS 302	EKMS Key Distribution Data Standard. National Security Agency, Director, National Security Agency, Ft. George G. Meade, MD. 20755-6734.
EKMS 311	EKMS ACCORDION 1.3 Length Indicator and Binding Code Specification. National Security Agency, Director, National Security Agency, Ft. George G. Meade MD. 20755-6734

This table is (U//FOUO)

Name	Description
EKMS 603	Interface Specification for the Data Transfer Device AN/CYZ-10. National Security Agency, Director, National Security Agency, Ft. George G. Meade, MD. 20755-6734
W3C Standards	
XAdES	J.C. Cruellas, G. Karlinger, K. Sankar XML Advanced Electronic Signatures; W3C Note 20 February, 2003 http://www.w3.org/TR/XAdES/
XML	Bray, T., Paoli, J., Sperberg-McQueen, C. M., Maler, E., "Extensible Markup Language (XML) 1.0 (Second Edition)," W3C Recommendation 6 October, 2000
XMLENC	Eastlake, D., Reagle, J., Imamura, T., Dillaway, B., Simon, E., "XML Encryption Syntax and Processing," W3C Recommendation 10 December, 2002
XMLSIG	Eastlake, D., Reagle, J., Solo D., "(Extensible Markup Language) XML-Signature Syntax and Processing," RFC 3075, March, 2002
XMLSEC	Mactaggart, M., "Enabling XML Security: An introduction to XML encryption and XML signature," http://www-106.ibm.com/developerworks/xml/library/s-xmlsec.html/index.html
KMI-2200	July, 2004

This table is (U//FOUO)

2.7.3.3.5 (U) Dependencies

(U//FOUO) The success of KM technologies depends on the successful specification and completion of the new and improved emerging infrastructures, mainly KMI.

2.7.3.3.6 (U) Complementary Technologies

(U//FOUO) KMI attempts to encompass a number of complementary (but disparate) technologies found in PKI, CMCS (COMSEC Material Control System), and EKMS.

(U) CKM complements PKIs by adding the Authorization component and works with all the leading PKI technologies.

2.7.3.4 (U) Certificate Management

2.7.3.4.1 (U) Technical Detail

2.7.3.4.1.1 (U) Certificate-Managed Infrastructures

(U) Certificate Management ties closely with Key Management. Certificates are widely used today within the PKI, based on the ANSI X.509v3 standards. Certificates use asymmetric keys, which are public/private key pairs.

(U) There are three independent PKI infrastructures that manage certificates today. These certificate management infrastructures are also addressed in the KMI Architecture and vision, as detailed in the section on Key Management. Figure 2.7-8 shows the three independent certificate management infrastructures that exist today in the government and commercial arenas.

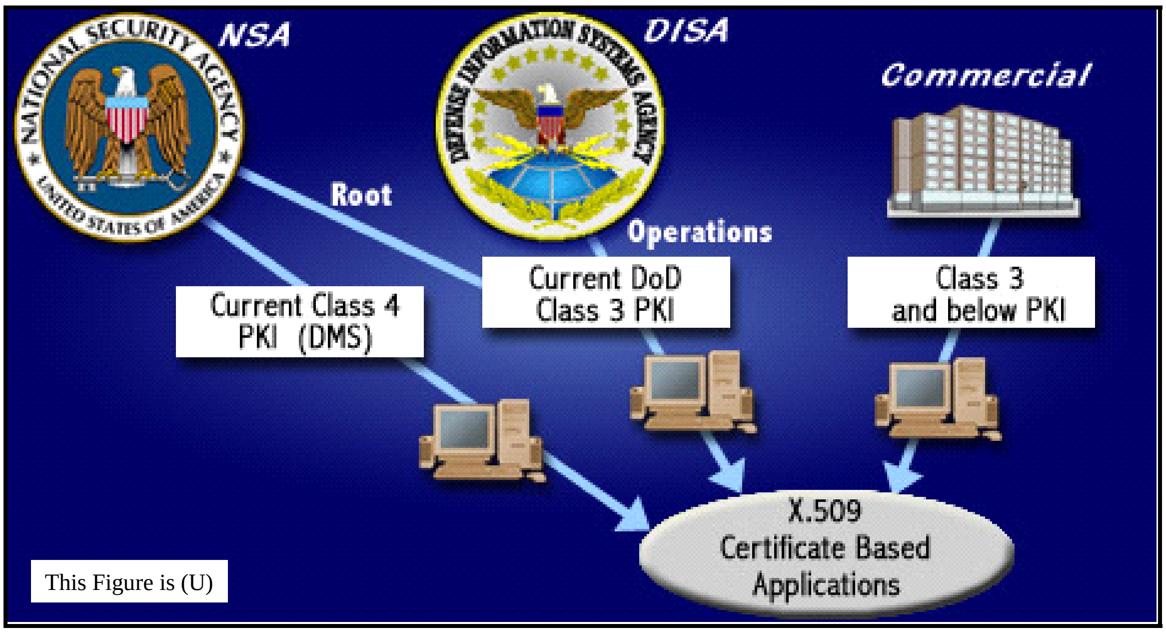


Figure 2.7-8: (U) DoD and Commercial Certificate-Managed Infrastructures

2.7.3.4.1.2 (U) Certificate Assurance levels

(U) The DoD specifies assurance levels for certificates, and technologies have been built to these Certificate level specifications. Overall, there are three Certificate Assurance levels in the DoD PKI. These are Class 5, Class 4 and Class 3 certificates. Class 5 is still under development, but technologies based on Classes 4 and 3 are operational today in various forms, and a brief description of these classes of certificates follow.

(U) Class 4 (Operational — based on NSA Technology)
(U//FOUO) The Class 4 PKI serves to protect Sensitive But Unclassified (SBU) information for the Defense Message System (DMS). It uses the FORTEZZA card as the user token for the storage of the Private key. It is designed to manage SBU and Secret information. It contains an individual's private key and the cryptographic algorithms for encryption and digital signature. FORTEZZA Plus card is primarily used with the STE. It is designed to protect information up to and including Top Secret.

(U) The Certificate Authority Workstation generates and manages certificates within the Government Class 4 PKI.

(U) Class 3 (Operational - Based on Commercial technology)
DoD Class 3 PKI: This PKI serves to protect mission critical information, and provides mission and administrative support. NSA serves as the Root CA and the Defense Information Systems Agency (DISA) manages operations. Private keys are stored on software tokens, such as floppy disks.

(U) The last PKI used by the Government is Commercial based. It is not controlled or operated by the Government. Certificates and keys are entirely generated and managed within the private sector. Private keys are generally on a software token. A private company serves as the Root CA. The PKI structure enables the Government to participate in E-Commerce activities.

2.7.3.4.1.3 (U) PKI Technology Model

(U) The PKI Technology Model illustrated in Figure 2.7-9 divides the PKI landscape into five layers and expresses requirements in terms of this model.

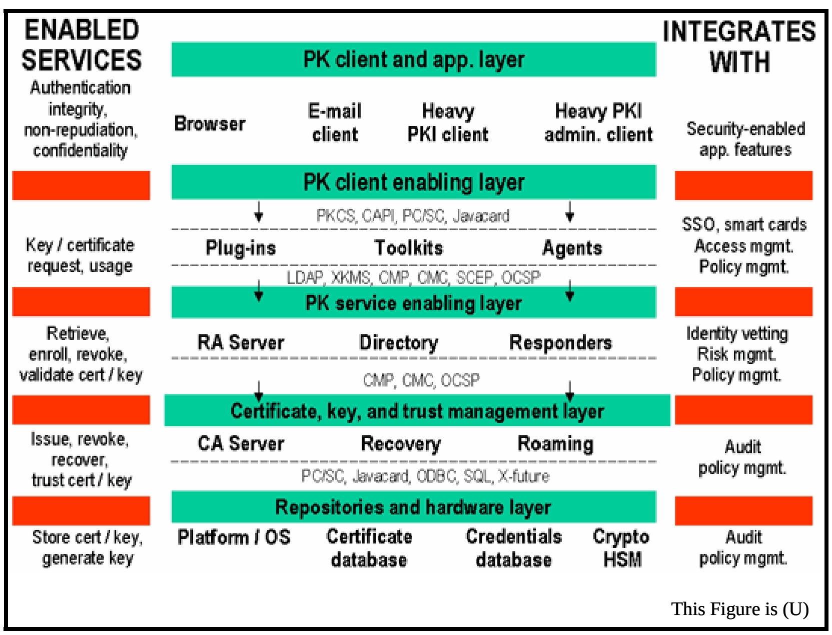


Figure 2.7-9: (U) PKI Technology Model

2.7.3.4.1.3.1 (U) Client and Application Layer

(U) Native PK-enabled browser, micro-browser, e-mail, and VPN clients in most cases are able to work with RAs and other enrollment gateways from the PKI products to obtain, check, and use certificates. There is the need for plug-ins to enhance security functionality in commodity browsers and e-mail programs. In some cases, there is a need for full desktop security clients, to enable functions such as file encryption, smart card support, secure e-mail with key history, dual encryption and signature key pair support, and CRL/OCSP checking usually accomplished by replacing an entire browser-based trust model with a more scalable and policy-managed PKI client implementation. The need for heavy PKI clients is declining, however, with improved CRL checking, dual key, smart card support and other features in the latest Internet Explorer (IE) and Communicator browsers and higher, as well as native file encryption in Windows XP. In fact, recent updates to IE have activated certificate validation, causing VeriSign to implement additional servers to handle the extra requests.

(U) Also, the client is the focus for administration of PKI systems. Some vendors implement a Windows or UNIX-based management client to administer their core RA and CA infrastructures, while others provide browser-based management, and some do both.

2.7.3.4.1.3.2 (U) Client Enabling Layer

(U) Whenever an application does not natively support PKI, vendors might provide plug-ins, toolkits, applets, or agents to help. These client enablers should preferably operate in conjunction with platform-resident security APIs such as Microsoft's Crypto API or in a stand-alone manner. Java applets or servlets let vendors immaculately transplant PK functionality into an application—no code installation required. Plug-ins are most effective in enhancing PK functions of mass market browsers and other clients with existing security hooks, but do leave a code footprint behind. Toolkits and APIs are required for custom integration of more obscure applications. Agents can enable certificate-equipped clients to sign onto other security domains through SSO or portal systems.

2.7.3.4.1.3.3 (U) Service Enabling Layer

(U) Enabling services include RAs, directories, and PK responders that a field client requests for enrollment, retrieval, recovery, validation, roaming, and other services. Ideally, they operate through standard protocols such as CMP, OCSP, SCEP, and XKMS. Directories are essential, as they allow clients to retrieve certificates, check policies, and check CRLs using LDAP. RAs take enrollment, recovery and revocation requests, vet them, and pass them on to the certificate infrastructure. Sometimes the RA function should be interactive or in other cases automated, particularly for enrollment. RAs may receive many batched user enrollment requests, and issue or deny certificates to those users based on rules in an automated identity vetting system.

(U) RAs may serve as gateways implementing protocols, such as SCEP, for automated pickup of certificates by machines or application services. For example, Windows XP and Server 2003 enable auto enrollment of machines and users.

(U) There is a push for the service-enabling layer to help thin out the PKI client in order to reduce the burden on application developers. PK responders implementing OCSP V2 and XKMS must supplement and replace today's simple OCSP V1 responders. Validation must become more sophisticated, linking with policy management and automated risk management systems provided by credit bureaus and other businesses.

2.7.3.4.1.3.4 (U) Certificate, Key, and Trust Management Layer

(U) Certificate servers (or CAs) must obtain their root credentials, enter into trust relationships by signing cross certificates or certificates of subordinate CAs, and issue and revoke client certificates. Recovery servers allow clients to obtain backup copies of private keys and certificates. Roaming servers provide securely stored credentials on demand to properly authenticated users. Note that PKI vendors implement and package certificate, recovery, and roaming servers in different ways.

2.7.3.4.1.3.5 (U) Repositories and Hardware Layer

(U) No PKI system would be whole without rock-solid underlying computer platforms, databases, and hardware security modules (HSMs) provided by best-in-class vendors. Customers must locate CAs, recovery, and roaming servers on UNIX, Windows 2000, Windows Server 2003, or other OS/hardware platforms as securely as possible. The PKI servers must also leverage robust databases with strong performance, backup, and audit features. Finally, it should be possible to store CA root keys and archived private keys in HSMs. In fact, CAs may depend on HSMs to achieve the FIPS 140-1 or ITSEC compliance levels needed for government certification or certification from private organizations like Identrus. HSMs can also accelerate signing, signature checking, and encryption processing performance. HSMs, CAs, and applications must implement common asymmetric, symmetric, and message digest crypto algorithms.

2.7.3.4.1.3.6 (U) Wireless Considerations

(U) Wireless PKI functionality is similar to wired PKI functionally, though requiring support for different product components such as wireless software toolkits, PKI Portal RAs, and CAs capable of issuing WTLS certificates. PKI systems must support short-lived certificates where micro-browsers cannot validate certificates online or store root keys. PKI vendors offering outsourced services need to get their root certificates implanted in devices just as they have done in browsers.

2.7.3.4.2 (U) Usage Considerations

2.7.3.4.2.1 (U) Implementation Issues

(U) Interoperability of components among multi-vendors is a critically important issue for infrastructures that support key and certificate management. Interoperability is used to describe the ability for one application to communicate seamlessly with another. Other aspects of interoperability include the ability to mix and match various PKI components from various vendors. Interoperability can also refer to the interaction between one enterprise domain and another (e.g., in order to conduct secure business-to-business transactions). Interoperability would allow greater flexibility and freedom of choice between vendor solutions and lowers the risk of deploying a PKI-based solution.

(U) The lack of interoperability is perceived as the leading barrier to wide-scale deployment of PKIs. Indeed, one of the fundamental reasons for the formation of the PKI Forum in December, 1999, was to identify and resolve existing barriers to multi-vendor interoperability.

(U) The PKI Forum (http://www.pkiforum.org/pdfs/PKIInteroperabilityFramework.pdf) has identified three major interoperability areas that require enhancements:

- (U) Component-Level Interoperability
- (U) Application-Level Interoperability
- (U) Inter-Domain Interoperability

2.7.3.4.2.2 (U) Advantages

(U) The PKI infrastructure and public certificates have been around for many years. One of its advantages is longevity. There have been many improvements along the way, but there are still challenges ahead.

2.7.3.4.2.3 (U) Risks/Threats/Attacks

(U) There are two primary entities vulnerable to attack are the subscriber and the CA. When there is a subscriber compromise, all subscribers within the entire infrastructure can be exploited until the compromise is detected. If there are no subordinate CAs, and the Root CA was compromised, the entire PKI could be compromised with devastating results. New keys and certificates would be required through the entire infrastructure. If a subordinate CA is compromised, only that CA and its subscribers must initiate actions to recover. This would still be an enormous amount of work, which is the reason extreme measures are required to protect the CAs.

(U) The CA must therefore maintain the integrity of its operations. The policies and procedures for its operation must be strictly adhered to at all times. Compromises of individual subscribers must be quickly and efficiently remedied and new keys generated, as appropriate. Concurrently, the Compromised Key List would need to be updated. Should the CA itself be compromised, all CA subscribers would need to be rekeyed and new Certificates created.

2.7.3.4.3 (U) Maturity

(U) The Infrastructure of public certificates, i.e., the PKI, has been around for many years, and as such has undergone significant growth and maturity. The maturity of this technology is rated as Emerging (TRLs 4 - 6). However due to the lack of interoperability standards for technologies within the infrastructure and the lack of security policy mandates, there is still reluctance for enterprises with need for high assurance to adopt the PKI standard. The maturity of Certificate Management is also rated as Emerging (TRLs 4 - 6).

2.7.3.4.4 (U) Standards

(U) Table 2.7-5 highlights some of the components and the standards with which PKI products comply.

13223 **Table 2.7-5 (U) Key Management and Certificate Management Standards**

This Table is (U)

Name	Description
Symmetric Encryption Algorithms	
DES	U.S. Data Encryption Standard (DES) in accordance with U.S. FIPS PUB 46-2 and ANSI X3.92
AES	U.S. Advanced Encryption Standard (AES) in accordance with U.S. FIPS PUB 197 (256-bit keys supported)
CAST block cipher	CAST block cipher in accordance with RFC 2144 (64-bit, 80-bit, and 128-bit variations are supported)
Triple-DES	Triple-DES in accordance with ANSI X9.52 (3-key variant for an effective key size of 168-bits is supported)
RC2®	RC2® in accordance with RFC 2268 (40-bit and 128-bit variations are supported)
IDEA	IDEA as listed in the ISO/IEC 9979 Register of Cryptographic Algorithms (128-bit supported)
Note: DES, CAST, Triple-DES, RC2 and IDEA encryption all use CBC mode of operation in accordance with U.S. FIPS PUB 81, ANSI X3.106 and ISO/IEC 10116	
Digital Signature Algorithms	
RSA	RSA in accordance with Public Key Cryptographic Standards (PKCS) specification PKCS#1 Version 2.0, ANSI X9.31, IEEE 1363, ISO/IEC 14888-3 and U.S. FIPS PUB 186-2 (1024-bit, 2048-bit, 4096-bit and 6144-bit supported)
DSA	DSA in accordance with the Digital Signature Standard, U.S. FIPS PUB 186-2, ANSI X9.30 Part 1, IEEE P1363 and ISO/IEC 14888-3 (1024-bit supported)
ECDSA	ECDSA in accordance with ANSI X9.62, IEEE P1363, ISO/IEC 14888-3 and U.S. FIPS PUB 186-2 (192-bit default)
One-Way Hash Functions	
SHA-1, SHA-256, SHA-384 and SHA-512	SHA-1, SHA-256, SHA-384 and SHA-512 in accordance to U.S. FIPS PUB 180-2 and ANSI X9.30 Part 2
MD5 Message-Digest algorithm	MD5 Message-Digest algorithm in accordance with RFC 1321
MD2 Message-Digest algorithm	MD2 Message-Digest algorithm in accordance with RFC 1319
RIPEMD-160	RIPEMD-160 in accordance with ISO/IEC 10118-3:1998
Key Exchange Algorithms	
RSA key transfer	RSA key transfer in accordance with RFC 1421 and RFC 1423 (PEM), PKCS#1 Version 2.0, IEEE P1363

This Table is (U)	
Name	**Description**
Diffie-Hellman key agreement	Diffie-Hellman key agreement in accordance with PKCS#3
Simple Public-Key GSS-API Mechanism (SPKM) authentication and key	Simple Public-Key GSS-API Mechanism (SPKM) authentication and key agreement in accordance with RFC 2025, ISO/IEC 9798-3 and U.S. FIPS PUB 196
SSL v3 and TLS v1	SSL v3 and TLS v1 in accordance with RFC 2246
Symmetric Integrity Techniques	
MAC	MAC in accordance with U.S. FIPS PUB 113 (for DES-MAC) and X9.19
HMAC	HMAC in accordance with RFC 2104
Pseudo-Random Number Generator	
Pseudo random number generator	Pseudo random number generator in accordance with ANSI X9.17 (Appendix C) and FIPS 186-2
Certificates and Certificate Revocation Lists (CRLs)	
Version 3 public-key certificates and Version 2 CRLs	Version 3 public-key certificates and Version 2 CRLs in accordance with ITU-T X.509 Recommendation and ISO/IEC 9594-8 (4th edition, 2000 as well as earlier editions)
Version 3 public-key certificate and Version 2 CRL extensions	Version 3 public-key certificate and Version 2 CRL extensions in accordance with RFC 2459 and RFC 3280
Version 3 public-key certificate and Version 2 CRL extensions in accordance with U.S. FPKI X.509 Certificate and CRL Extensions Profile	Version 3 public-key certificate and Version 2 CRL extensions in accordance with U.S. FPKI X.509 Certificate and CRL Extensions Profile
Version 3 public-key certificate and Version 2 CRL extensions in accordance with NIST X.509 Certificate and CRL Extensions Profile for the Common Policy	Version 3 public-key certificate and Version 2 CRL extensions in accordance with NIST X.509 Certificate and CRL Extensions Profile for the Common Policy
Version 3 "Qualified" certificates in accordance with RFC 3039 and ETSI TS 101 862	Version 3 "Qualified" certificates in accordance with RFC 3039 and ETSI TS 101 862
Version 3 public-key certificates and Version 2 CRLs in accordance with de-facto standards for Web browsers and servers	Version 3 public-key certificates and Version 2 CRLs in accordance with de-facto standards for Web browsers and servers
WTLS Certificate support in accordance with WAP WTLS Version 1.1. (certificate issuance)	WTLS Certificate support in accordance with WAP WTLS Version 1.1. (certificate issuance)
RSA algorithm identifiers and public key formats in accordance with RFC 1422 and 1423 (PEM) and PKCS#1	RSA algorithm identifiers and public key formats in accordance with RFC 1422 and 1423 (PEM) and PKCS#1

This Table is (U)	
Name	**Description**
Online Certificate Status Protocol, version 2. Working document of the Internet Engineering Task Force (IETF) RFC 2560.	Online Certificate Status Protocol, version 2. Working document of the Internet Engineering Task Force (IETF) RFC 2560.
File Envelope Formats	
Standard file envelope format based on Internet RFC 1421 (PEM)	Standard file envelope format based on Internet RFC 1421 (PEM)
PKCS#7 Version 1.5 based on RFC 2315 and Cryptographic Message Syntax (CMS) based on RFC 3369 and 3370	PKCS#7 Version 1.5 based on RFC 2315 and Cryptographic Message Syntax (CMS) based on RFC 3369 and 3370
S/MIME Version 2 based on RFC 2311	S/MIME Version 2 based on RFC 2311
Secure Session Formats	
On-line GSS-API public key implementation mechanism using SPKM in accordance with Internet RFC 2025 and SPKM entity authentication in accordance with FIPS 196	On-line GSS-API public key implementation mechanism using SPKM in accordance with Internet RFC 2025 and SPKM entity authentication in accordance with FIPS 196
SSL v3 and TLS v1 in accordance with RFC 2246	SSL v3 and TLS v1 in accordance with RFC 2246
Repositories	
LDAP Version 2	LDAP Version 2 in accordance with RFC 1777 and RFC 2559
LDAP Version 3	LDAP Version 3 in accordance with RFC 2251-2256
Private Key Storage	
Private key storage	Private key storage in accordance with PKCS#5 and PKCS#8
Certificate Management	
Secure Exchange Protocol (SEP)	Secure Exchange Protocol (SEP), built using Generic Upper Layers Security (GULS) standards ITU-T Recs. X.830, X.831, X.832 and ISO/IEC 11586-1, 11586-2, 11586-3 (SEP continues to be supported for backward compatibility only)
PKIX-CMP	PKIX-CMP in accordance with RFC 2510 and PKIX-CRMF in accordance with RFC 2511
PKCS 7/10	PKCS 7/10 (for Web based clients and VPN solutions)
Cisco Certificate Enrollment Protocol (CEP)	Cisco Certificate Enrollment Protocol (CEP) (for VPN solutions)

This Table is (U)	
Name	**Description**
Application Programming Interfaces (APIs)	
Hardware cryptographic interface	Hardware cryptographic interface in accordance with PKCS#11
Generic Security Services API (GSS-API)	Generic Security Services API (GSS-API) in accordance with RFC 1508 and 1509
IDUP-GSS-API	IDUP-GSS-API in accordance with Internet Draft draft-ietf-cat-idup-gss-08.txt
This Table is (U)	

2.7.3.4.5 (U) Dependencies

(U) There needs to be component interoperability, application interoperability, and inter-organization interoperability. While PKI may never just disappear into the infrastructure, it should be reduced to a set of simpler, better-understood services and decisions. Furthermore, there is a need for integration with the OS, wireless, and smart card platforms as well as platform-neutral Java and XML functionality. Emerging Web Services, SAML, and other XML security specifications will benefit if PKI can be easily integrated. Both infrastructure and application vendors should give high priority to XKMS development (once the standard stabilizes and is ratified by W3C) to increase interoperability by thinning out the client layer of PKI and support WS-Sec, SAML, Liberty Alliance, XML Access Control Markup Language (XACML), Extensible Rights Markup Language (XrML), and other XML security specifications.

(U) Infrastructure vendors are preparing for a gradual evolution from PKIX toward XML-based PKI, shedding the ASN.1 heritage of OSI in favor of a universal text encoding. But this presumes that PKIX will eventually re-map X.509v3 certificates to XML encoding, and until then, there will be an ongoing need to preserve PKIX interoperability by implementing XKMS, CMP, CMC, and OCSP V2 (once it stabilizes) to achieve the broadest functionality and component interoperability.

(U) XML security standards are still quite immature and will require several more years before a broad suite is available for deployment in commercial products. But architects and planners can target a model architecture to leverage WS-Security. And federated identity will be ready to move as vendor software is available. In the meantime, vendors are developing WS-Sec and federated identity best practices that easily integrate PKI.

(U) Liberty Alliance circles of trust may provide a driver for enterprises to cross certify. Vendors must continue to engage these organizations. But no one consortium or trust network will unlock the real potential of PKI unless it helps users meet the need for mutual certificate acceptance by cross-certifying with others.

(U) Cross-certifying requires simpler, more compatible policies, for it is the policy that cleaves CAs apart by limiting which certificates users and organizations can trust. Industry consortiums and vendors alike should invest significant effort in projects such as the Federal Public Key Infrastructure (FPKI) Group's Bridge CA program, which pushes the boundaries of PKI with its effort to extend inter-organization interoperability by refining path processing, policy mapping, cross certification, and directory services between agencies and commercial organizations. Sites must be able to leverage pre-existing certificates.

2.7.3.4.6 (U) Alternatives

(U) There are no real alternatives to Certificate Management technologies. As indicated earlier, there is a drive to establish interoperability standards, such that components from various certificate management providers can interoperate.

2.7.3.4.7 (U) Complementary Technologies

(U) CKM and Key Management technologies—especially asymmetric key methodologies—complement the incorporation of Certificate Management infrastructures and technologies.

(U) XKMS defines a SOAP/XML-messaging-based alternative to traditional PKI, though in many ways XKMS is designed to complement, rather than replace, established PKI standards. At the client level, XKMS defines mechanisms under which applications delegate the retrieval, parsing, and validation of X.509 digital certificates to trusted servers, thereby streamlining the configuration of client-side trust-service business logic. XKMS requires retrofitting today's clients and applications to support, at a minimum, such standards as SOAP, XML-DSig, XML Schemas, XML Namespaces, WSDL, and XML Encryption.

2.7.3.5 (U) Configuration Management of IA Devices and Software

2.7.3.5.1 (U) Technical Detail

(U) The purpose of configuration management is to establish and maintain the integrity of IA components—hardware, firmware, and software throughout their life cycle. Configuration management involves identifying the configuration, controlling configuration changes, and maintaining the integrity and traceability of the configuration throughout the component's life cycle. Given the assured, dynamic, decentralized nature of the GIG, configuration establishment and control must be assured—only authorized authorities should be able to modify configurations. It must be remotely accessible, since GIG assets may be literally anywhere and configuration updates must be possible in the field without local manual intervention. It must also be auditable—there must be a mechanism for verifying a configuration is still valid.

(U) Configuration Items that must be securely managed within the GIG include such items as:

- (U) Operating System Software, particularly for trusted or high-assurance components
- (U) Router tables
- (U) Firewall configurations
- (U) VPN configurations
- (U) NDS configurations
- (U) Host-based IDS/IPS agent configurations
- (U) Malware detection and prevention agents, software and signature configuration files
- (U) CDS configurations
- (U//FOUO) Cryptographic modules and algorithms (hardware and software)
- (U) Keys and Certificates
- (U) Trusted applications.

(U) Some of these may be represented in hardware, firmware, or software. Many will require constant, regular updates to accommodate dynamic changes in the GIG and to fix discovered vulnerabilities or defects. Some, such as keys and certificates, require that strict accountability be maintained for their possession and distribution. Such items require packaging for distribution, receipts, and auditable tracking of any transactions. Management operations that must be performed include:

- (U) Maintaining the set of authorized configuration baselines
- (U) Installing a software configuration baseline
- (U) Provisioning a system—installing optional or additional software components according to the mission requirements for the target system.

- (U) Verifying the completeness and integrity of a software configuration in IA components against a baseline
- (U) Determining if upgrades or patches are necessary for an IA component
- (U) Upgrading software or installing patches
- (U) Installing and Upgrading third-party software applications
- (U//FOUO) Transferring, receipting and installing data packages
- (U) Reporting on the version and status of any IA component firmware or software including OS, system software, application software, and versioned data

(U) Such tasks as determining if upgrades or patches are necessary overlap with tasks such as vulnerability assessment, discussed in Section 2.6, Network Defense and Situational Awareness.

(U) A number of CM problems within the GIG already have point solutions, which are discussed below.

2.7.3.5.1.1 (U) Systems Management Applications

(U) Systems Management Consoles are centralized, dedicated systems that can manage other systems within an enterprise. They interact with the managed systems or clients through an installed agent. They can perform a variety of configuration management tasks using a proprietary communications protocol, which is highly extensible to allow development of additional operations on the clients. Actions that such servers can perform are:

- (U) Installation of the operating system remotely on a bare metal system for supported clients
- (U) Installation of data and applications or provisioning of client systems
- (U) Distribution and installation of software updates or patches and tracking of which machines did and did not receive updates
- (U) Forced remote execution of software on clients to perform such actions as malware detection updates
- (U) Verification and auditing of client system software configuration and versions
- (U) Asset tracking.

(U) System Management applications interact with an agent residing on the client machine to perform their operations. Additional applications can be added via scripts and the API, but this can be a complex programming task with attendant development, testing, and deployment issues. These applications generally support common desktop and server operating systems with some supporting models of PDA. APIs and custom software development can extend high-end management frameworks to handle operations beyond that originally envisioned—or client targets. In cases where there is a large market, such as popular routers, third parties such as the router vendors have written plug-ins or interfaces to their proprietary management applications that connect to the large management applications.

2.7.3.5.1.2 (U) Network Boot Applications

(U) A wide variety of desktop and server computer systems are capable of booting an un-configured machine from a network server that is discovered at boot time. For Intel processor-based computers, the Intel Preboot eXecution Environment (PXE) [INTEL] specification defines an interface for booting from the network. Most RISC-based processors also have network boot capability by default. They depend upon such standard protocols as the Dynamic Host Control Protocol (DHCP) [DHCP97], Trivial File Transfer Protocol (TFTP), and the Boot Protocol (BOOTP). They can be used to dynamically boot a diskless client off a central server or as an initialization step that then loads a bootstrap kernel to load a complete system onto a local disk for subsequent use. Servers or systems management consoles can be configured to supply standardized OS images to booting PCs.

(U) However, the underlying protocols are unauthenticated and depend upon network broadcast and are suitable only for a trusted, benign LAN environment. Since any server can respond, and the clients cannot authenticate to a server, the security vulnerabilities have proven so great that this mechanism is only used in special cases. The Intel PXE specification includes a Boot Integrity Services (BIS) API, but this is not widely available, and for high-assurance requirements requires making modifications to the Boot ROM of a system.

2.7.3.5.1.3 (U) Malware Management

(U) Virus detection is one of the more mature areas of IA. Viruses were one of the earliest attacks on computer systems, emerging shortly after the initial widespread adoption of personal computers. Because most virus detection software was signature-based, update mechanisms were developed early and have evolved with communication technologies. Current malware detection agents can automatically update themselves securely from central servers—both signatures and the application software itself. A number of virus vendors have enterprise management servers, which will manage the client malware detection agents in a local enterprise. These managers can generally perform the following:

- (U) Signature (data) file or application update download (pull) from the vendor per policy
- (U) Signature and application update to clients (push) per policy
- (U) Configuration of scan and update policy
- (U) Tracking of client update status (last contact, last version)

- (U) Tracking of enrolled clients (machines with and without malware detection agents)
- (U) Reporting statistics and consolidation of alerts.

(U) With current products, only the malware detection agent vendor can provide the associated management solutions. The format and structure of signature files and updates are proprietary, as are the protocols used to perform the updates. As a consequence, no malware management system can manage third party agents, and if general enterprise security console applications are able to monitor the agents on a network, they cannot perform the configuration updates on those agents.

2.7.3.5.1.4 (U) ECU Update

(U) Recent models of cryptographic hardware such as the KG 235 and KG-240 can be securely managed by a manager device over the network. The manager is capable of performing the following tasks remotely on a KG-240:

- (U) Updating the system software
- (U) Updating cryptographic algorithms
- (U) Updating keys
- (U) Updating security policies.

(U) The protocol for managing the devices is proprietary and unique to the KG-240. Devices such as the KG-235 do provide SNMP interfaces on both the red and black sides, but they are limited to standard SNMP operations and do not provide configuration management capabilities.

2.7.3.5.1.5 (U) Patch Management Systems

(U) Patch Management Systems are software applications that are specifically designed to centralize the distribution of operating system and specific application patches within an enterprise. Some are agent-based, with small agent servers installed on monitored clients. Others do not require an agent on the client targets. They use only the built-in capabilities of the resident OS to provide the hook into the target system. Although a number of patch management solutions operate on multiple architectures and operating systems, all investigated products currently target only desktop and server systems and smaller devices that run Microsoft Windows CE. None handle embedded systems or arbitrary client architectures.

2.7.3.5.2 (U) Usage Considerations

2.7.3.5.2.1 (U) Implementation Issues

(U) Systems Management Applications are very large and complex systems. They require a large, full time staff to use and maintain. Although very flexible and extensible, it comes at the cost of software development with its associated development, testing, and deployment issues.

(U) Agentless Patch Management systems suffer from significant network traffic from server to target machines. In contrast, Agent-based patch management applications can use the on-device agent to locally scan the machine for individual file version and configuration information.

2.7.3.5.2.2 (U) Advantages

(U) All these tools centralize one or more aspects of configuration management. For large systems management applications, they can centrally control many common aspects of configuration management.

2.7.3.5.2.3 (U) Risks/Threats/Attacks

No applications provide mechanisms to validate version numbers or system configurations using techniques as MD5 hashes to verify that critical files are unchanged. Although mechanisms exist to authenticate management servers, clients are not authenticated and the transactions are not generally protected, so they are unsuitable for high assurance applications.

(U) Agent-based CM applications

(U) For all configuration management systems that do not use secure communications, the threat is that an adversary could spoof the management console and take control or install arbitrary software on a client. If the client is not authenticated, then an adversary can spoof the client and receive keys or other cryptographic material and possibly assume the identity of the spoofed client. The issue of a spoofed client identity is discussed further in Section 2.7.2.1, Identity Management.

(U) Agentless CM applications

(U) For configuration management tools such as patch managers that are agentless, they use alternate means of accessing information, such as Microsoft NetBIOS file sharing and administrator login. Typically these services cannot be available on a machine except in the most benign environments due to extreme vulnerability, so such applications cannot be used at all outside the local enclave.

2.7.3.5.3 (U) Maturity

(U) The maturity is high for individual point solutions for various parts of configuration management. All of the various technologies have examples of successfully deployed product solutions in commercial environments. So, the maturity of CM technology is rated as Mature (TRLs 7 - 9). However, none of the technologies meets GIG requirements such as the high assurance required to securely manage Information Assurance Components (IAC) across a lower-assurance network.

2.7.3.5.4 (U) Standards

(U) Standards related to Configuration Management are included in Table 2.7-6.

13439 **Table 2.7-6: (U) Configuration Management Standards**

This Table is (U)	
(U) Name	**(U) Description**
IETF Standards	
SNMPv3	The Simple Network Management Protocol, version 3 is the latest version of the IETF standard for managing network devices. Version 3 includes authentication and authorization, so is considered much more secure than previous versions. SNMP is widely implemented, but has some significant restrictions because of its very simple structure.
TFTP	The Trivial File Transfer Protocol (TFTP), as defined by IETF RFC 1350, is a very simple file transfer protocol that can be implemented in very small systems, such as firmware. It implements no authentication whatsoever and consequently is usable only in the most benign, protected environments.
DHCP	The Dynamic Host Control Protocol (DHCP) is defined by IETF RFC 2131 and modified by a host of other RFCs. It allows a machine, which at network initialization time does not know its own IP address, to request allocation of an IP address from a server and receive network configuration data sufficient to communicate on an IP network.
The Open Group Standards	
SM Spec	Signed Manifest Specification, The Open Group SM Spec Signed Manifest Specification, The Open Group, 1997. http://www.opengroup.org/pubs/catalog/c707.htm
DMTF Standards	
CIM	The Distributed Management Task Force (DMTF) originally developed the Common Information Model (CIM) to provide a data model for integrating management across SNMP, the Desktop Management Interface (DMI) (another part of WBEM), Common Management Information Protocol (CMIP or ISO 9596) (for telecom devices) and private applications. CIM is part of the DMTF's overall Web-based Enterprise Management (WBEM) initiative. WBEM includes CIM as the data definition, XML as the transport/encoding method, and HTTP as the access mechanism. CIM is an object-oriented data model for describing managed elements across the enterprise, including systems, networks, and applications. The CIM schema provides definitions for servers, desktops, peripherals, operating systems, applications, network components, users, and others along with details of each. One of the main functions CIM offers is the ability to define the associations between components. CIM's object-oriented approach makes it easier to track the relationships and interdependencies between managed objects. WBEM/CIM proponents promote this as a key advantage over SNMP.
WBEM	The Web-Based Enterprise Management (WBEM) standard is an initiative by the DMTF to develop a broader enterprise management structure than SNMP. The DMTF is an industry coalition that is developing an enterprise management framework for computer systems that is richer than SNMP

This Table is (U)	
(U) Name	**(U) Description**
SMBIOS	The System Management Basic I/O System (SMBIOS) is a DMTF standard for making firmware-level information available via a CIM model on computer systems.
Vendor Standards	
Intel PXE specification	The Intel-developed Preboot eXecution Environment (PXE) specification defines an OS-independent firmware-level mechanism for booting from a variety of media, including the network, using standard protocols. ftp://download.intel.com//labs/manage/wfm/download/pxespec.pd
Intel PXE BIS specification	The Intel PXE Boot Integrity Services is an extension to the Intel PXE specification that provides for PKI-based authentication of the server to the booting client. ftp://download.intel.com//labs/manage/wfm/download/bisspec.zip
This Table is (U)	

2.7.3.5.5 (U) Cost/Limitations

(U) Systems management applications can provide full management of client systems, but are extremely expensive—reaching $1000 per client system or more in annual licensing costs.

(U) All systems have no support for non-standard target machines. Although the general systems management applications can be extended to cover embedded systems or appliances, it is a custom software development. Specialized hardware such as IDS appliances, HAIPEs, or specialized military hardware with IA components are unsupported by any commercial implementation. Some applications can be extended to included non-standard clients, but this is only with custom software development.

2.7.3.5.6 (U) Dependencies

(U) Many of the current products assume a native patch management mechanism exists for the target machine such as the Microsoft Installer (MSI) for Microsoft Windows clients or something like Redhat Package Manager (RPM) for Linux clients, and either use it directly or develop a common proprietary packaging scheme that unpacks on the target machine into a native format. All of the configuration management tools depend upon the OS-native application version and configuration data to be correct and valid. None of the current products provide an independent server-based record of a client installation for comparison to the current configuration or validation of the contents of files.

2.7.3.5.7 (U) Complementary Techniques

(U) The determination of the optimal configuration of an IA device is intimately related to the vulnerabilities of that device and its associated software, so many configuration assessment tools are integrated with a general vulnerability assessment scanner, or they derive their configuration definition from a vulnerability assessment tool.

2.7.3.5.8 (U) References

(U) [SNMP02a] "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks," RFC 3411; D. Harrington, R. Presuhn, B. Wijnen, December, 2002. http://www.ietf.org/rfc/rfc3411.txt

(U) [SNMP02b] "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)," RFC 3412, J.D. Case, D. Harrington, R. Presuhn, B. Wijnen, December, 2002. http://www.ietf.org/rfc/rfc3412.txt

(U) [SNMP02c] "Simple Network Management Protocol (SNMP) Applications," RFC 3413. D. Levi, P. Meyer, B. Stewart. December, 2002. http://www.ietf.org/rfc/rfc3413.txt

(U) [SNMP02d] "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)," RFC 3414, U. Blumenthal, B. Wijnen. December, 2002. http://www.ietf.org/rfc/rfc3414.txt

(U) [SNMP02e] "Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)," RFC 3418, R. Presuhn, Ed.. December, 2002. http://www.ietf.org/rfc/rfc3418.txt

(U) [CIM99] "Common Information Management (CIM) Specification, DSP004, The Distributed Management Task Force, Inc. Version 2.2," June 14, 1999. http://www.dmtf.org/standards/documents/CIM/DSP0004.pdf

(U) [SMBIOS02] "System Management BIOS Specification, The Distributed Management Task Force, Inc. v2.3.4," DSP0134, December 6, 2002. http://www.dmtf.org/standards/documents/SMBIOS/DSP0134.pdf

(U) [TFTP] "The TFTP Protocol," RFC 1350, Sollins, K. July, 1992. ftp://ftp.rfc-editor.org/in-notes/rfc1350.txt

(U) [DHCP97] "Dynamic Host Configuration Protocol," RFC 2131, R. Droms, March, 1997. http//www.ietf.org/rfc/rfc2131.txt

(U) [WBEM] "WBEM Discovery using SLP, DSP0205," The Distributed Management Task Force, Inc. Version 1.0.0, Jan. 27, 2004. http://www.dmtf.org/standards/wbem/DSP0205.pdf

(U) [INTEL] Intel-developed Preboot eXecution Environment (PXE) specification. ftp://download.intel.com//labs/manage/wfm/download/pxespec.pdf

(U) Intel PXE BIS specification
(U) ftp://download.intel.com//labs/manage/wfm/download/bisspec.zip

(U) [CMS02] "Cryptographic Message Syntax," RFC3369, R. Housley, August, 2002, http://www.ietf.org/rfc/rfc3369.txt.

(U) "Symantec Enterprise Security Architecture (SESA™)," Symantec Corporation, 2002. itpapers.zdnet.com/abstract.aspx?scid=284& tag=tu.sc.ont.dir3&x=80&docid=87493

(U) "Virus and Vulnerability Classification Schemes: Standards and Integration," S. Gordon, Symantec. 2003. http:// securityresponse.symantec.com/ avcenter/reference/virus.and.vulnerability.pdf

2.7.3.6 (U) Inventory Management

2.7.3.6.1 (U) Technical Detail

(U) A key element to managing GIG assets is an ability to dynamically create and maintain an accurate inventory of IA assets. There are three components to an automated inventory management system—the data entry mechanism, central database, and reporting system. An emerging technology to support data entry and collection is Radio-Frequency Identification (RFID). RFID is a technology that offers the ability to add small radio transponders to objects that respond to an RF signal with a small amount of information. With advances in manufacturing technology, RFID tags are now small enough to be embedded in banknotes and are rapidly become sufficiently inexpensive to attach to relatively inexpensive items, which enables a large number of widespread inventory, supply-chain, and tracking and identification applications. For a number of logistics applications the DoD is currently piloting RFID tags, and USD/ATL has issued a policy memorandum [DOD04] specifying use of RFID tags for large classes of logistics applications by January 1, 2005. They have significant advantages over other approaches for inventory tagging:

- (U) No physical contact or line of sight is required, only proximity—removal from packaging is not a requirement
- (U) They are relatively immune to dirt, chemicals, or temperature variations
- (U) Many RFID tags can be read virtually simultaneously. This yields scan rates much higher than barcodes that require manual scanning of each individual barcodes

(U) An RFID system is composed of three components:

- (U) Tag
- (U) Antenna
- (U) Reader

(U) The tag is a small electrical device that is—at its simplest— silicon chip connected to an antenna. Other forms include a smart label or a rectangular case.

(U) The reader is a device that reads RFID tags. There are many varieties, from small hand-held devices to fixed readers for smart shelves or warehouse doorways. They may have integral antennas or separately attached antennas. Readers are placed at key locations where they can track tags as they pass automatically, such as in warehouse doorways, loading docks, and inspection points. Emerging applications are smart shelves that can report their contents automatically and readers on forklifts that automatically identify when the correct pallet is being lifted or moved.

(U) An example application is shown in Figure 2.7-10. The central inventory application is what stores and processes the data from the reader. It can reside anywhere on the GIG, but it must be accessible by the reader hardware and software.

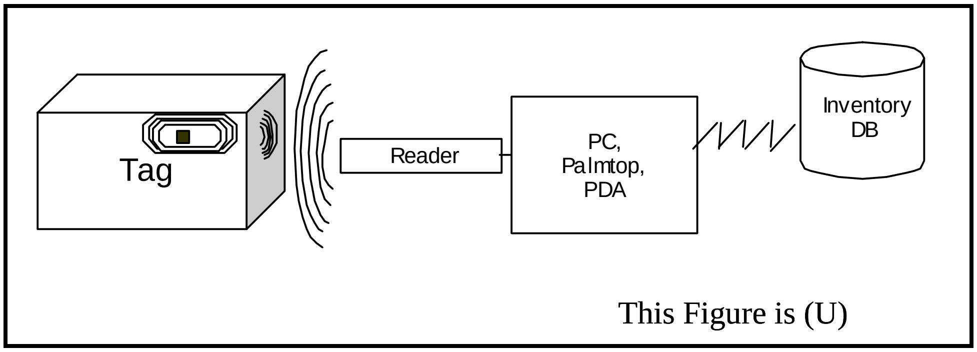


Figure 2.7-10: (U) RFID Operation

(U) RFID tags come in three major varieties:

- (U) Active RFID tags consist of an antenna, transponder chip, and a power source such as a battery. They have a lifetime limited by the battery charge
- (U) Passive RFID tags have no power source, but are powered solely by the RF energy of the external reader
- (U) Semi-passive RFID tags have a power source to improve performance, but the power is used solely to power the internal circuitry during operation and is not used to generate RF signals. They are intermediate in cost and capability.

(U) RFID tags can be manufactured in a variety of form factors. With printed or etched antennas and single-chip transponders, they can be manufactured as adhesive labels that can be read without physical contact. Range and data rate performance of RFID tags varies widely depending upon the type of tag and the environment. The minimum for passive tag ranges are a few inches and a capacity of 64 or 96 bits. Active tags can reach up to 100 ft with up to 2Kb data. Emerging UHF-band passive tags have longer ranges.

(U) The most inexpensive tags are read-only, set at manufacture. Other tags can be programmed once with an ID code. Read-write tags can store mutable information in addition to a fixed serial number.

2.7.3.6.2 (U) Usage Considerations

2.7.3.6.2.1 (U) Implementation Issues

(U) RFID tags have some significant physical limitations, primarily in range. Passive tags must have close physical proximity to the reader to receive a strong enough signal to energize the circuit enough to send a detectable response (the signal strength varies as the fourth power of the distance). For close range devices, this is accomplished by making the reader a hand-held scanner, which then uses a conventional wireless communications technology, such as 802.11b, for communications with the network and central database server.

(U) As RF devices, RFID tags are affected by the same environmental considerations common to all RF devices. Metal or conductive objects block RF signals. The antenna must be outside and physically separated from a metal enclosure. Otherwise it acts a Faraday cage[13]—blocking all signals. RFID tags are subject to interference from other RF sources, electrical equipment, or motors. In addition, in many environments such as loading docks, readers may interfere with each other, reading tags in other docks, requiring additional signal processing to avoid ambiguity or errors.

(U) Although RFID tags individually have low bandwidth requirements, when processing large numbers of tags, the readers may use significant bandwidth to communicate with the host application in real time. Hence, large amounts of equipment processing in low-bandwidth tactical communications environments might use a significant amount of bandwidth.

(U) Currently, RFID tags operate in regions of the frequency spectrum reserved for Industrial, Scientific, and Medical (ISM) Applications, as detailed in Table 2.7-7.

Table 2.7-7: (U) Frequency Ranges for RFID Systems

This Table is (U)

Frequency Range	Comment	Allowed Field Strength / Transmission Power
< 135 kHz	Low frequency, inductive coupling	72 dBµA/m
6.765 - 6.795 MHz	Medium frequency (ISM), inductive coupling	42 dBµA/m
7.400 - 8.800 MHz	Medium frequency, used for EAS (electronic article surveillance) only	9 dBµA/m
13.553 - 13.567 MHz	Medium frequency (13.56 MHz, ISM), inductive coupling, wide spread usage for contactless smartcards (ISO 14443, MIFARE, LEGIC,), smartlabels (ISO 15693, Tag-It, I-Code,) and item management (ISO 18000-3).	42 dBµA/m
26.957 - 27.283 MHz	Medium frequency (ISM), inductive coupling, special applications only	42 dBµA/m
433 MHz	UHF (ISM), backscatter coupling, rarely used for RFID	10 .. 100 mW
868 - 870 MHz	UHF (SRD), backscatter coupling, new frequency, systems under development	500 mW, Europe only
902 - 928 MHz	UHF (SRD), backscatter coupling, several systems	4 W - spread spectrum, USA/Canada only
950 - 956 MHz	UHF (SRD), backscatter coupling, new frequency	Power TBD, Japan only
2.400 - 2.483 GHz	SHF (ISM), backscatter coupling, several systems, (vehicle identification: 2.446 .. 2.454 GHz)	4 W - spread spectrum, USA/Canada only, 500 mW, Europe
5.725 - 5.875 GHz	SHF (ISM), backscatter coupling, rarely used for RFID	4 W USA/Canada, 500 mW Europe

This table is (U//FOUO)

[13] (U) A Faraday cage is any conductive surface which surrounds an antenna. Any electromagnetic field is canceled inside a conductor, so no RF can ever pass through.

(U) As shown, the emerging, UHF RFID spectrum is different for the United States, Europe, and Japan. This makes a common worldwide solution more challenging.

2.7.3.6.2.2 (U) Advantages

(U) RFID tags offer the ability to reliably process large numbers of IA components just by physical proximity. The ability to track pallet loads of devices automatically, merely by moving them through the warehouse door with no data entry error, represents a significant improvement in tracking. Smart shelves that know what items are stored on them and that can communicate to an inventory application can revolutionize inventory management.

2.7.3.6.2.3 (U) Risks/Threats/Attacks

(U) RFID tags are RF transponders that respond whenever they are probed. With an absolute minimum of circuitry for power and cost reasons, they contain no circuitry capable of supporting complex encryption, decryption, or authentication operations. Passive smart label RFID chips contain only enough circuitry to broadcast a 64- or 96-bit serial number. Although the RFID tag information itself would rarely be classified, to be useful, it must be connected to status and descriptive information for the component, which may be classified.

(U) The third component of any RFID system is the host application. For IA component inventory and tracking applications, this will certainly involve sensitive or classified information such as current keysets and algorithms. As a result, either the database must operate in a multiple security domain configuration or the reader and all communications links must be capable of operating at the required assurance and confidentiality levels. Commercial hand-held RFID readers which use 802.11b/g for communications with the host application do not support the level of protection required for such information. Many offer applications which display the status of any component scanned on a local screen. When the inventory item is a high-value sensitive IA component or an element of a larger such component, communication with the centralized database becomes sensitive or classified.

(U) In addition, there are a number of attacks that are possible with RFID systems:

- (U//FOUO) Attack – Unauthorized Read Tag. An attacker can determine the inventory of sensitive equipment simply by using a commercial RFID reader, perhaps with an extended-range antenna to query the RFID tags in the same manner as an authorized user. This could present a very significant vulnerability in a battlefield or tactical environment where every tag represents the equivalent of a IFF transponder broadcasting a location. Tags are currently being developed which can be “deactivated” upon command, but they are primarily being developed in response to consumer privacy concerns, not authentication concerns, so the potential deactivation operations are permanent and non-reversible. More complex tags that allow soft deactivation and reactivation are being developed, but the cost will be significantly higher, and they will not have any authentication features.

- (U//FOUO) Attack – Remove tag or cover tag – Tags which are mounted externally for shielding and range also become vulnerable to removal from the equipment, which in an automated environment would cause it to disappear from inventory and tracking. A similar result can be achieved with foil or a wire mesh covering the antenna.

- (U//FOUO) Attack – Replace tag ID information – More sophisticated RFID tags that have read-write capability will rewrite their data on any command from any RFID reader. No authentication is available. A handheld reader can transform a high-value sensitive piece of equipment into an innocent, low-value item for easy removal from the warehouse.

2.7.3.6.3 (U) Maturity

(U) Although RFID tags have existed since 1974, only within the last few years has the price of tags dropped to the level that makes them feasible for wide-scale deployment within the supply chain infrastructure. The DoD has issued and updated an RFID policy mandating the use of RFID tags for certain shipping containers and large pallet-sized shipments by Jan 1, 2005, with further expansion of use over the next few years. UHF tags, which appear to have the greatest promise for low-cost, long-range usage—ideal for inventory applications—are just now being developed by manufacturers and are not in widespread use. No readers currently operate at all three (U.S., European, and Japanese) UHF bands. The current drive is to reduce tag manufacturing costs, so security enhanced tag systems may be some time in coming.

(U//FOUO) A key element of RFID for GIG inventory management is that the RFID tags must be secure. Many IA assets will be used in combat, and inadvertent or adversary-triggered RF transmissions from RFID tags would be a serious vulnerability. A key enhancement would be the ability to activate and deactivate tags before and after missions. A greater issue is that current RFID tags have no authentication or authorization capability at all. Any reader can interrogate a tag, and any reader can write or rewrite writeable tags. With extremely limited on-board processing capacity, the capacity to restrict functions to authenticated, authorized readers is a number of years away.

(U) The maturity of tag technology for general inventory management is rated as Emerging (TRL 4-6). There are large-scale DoD and commercial pilot programs underway, such as those initiated by Walmart and Gillette. However, current pilot programs are not addressing secure RFID tags for assured inventory management, and significant vulnerabilities of conventional tags have not been addressed. Accordingly, maturity of RFID technology that would meet the security requirements of the GIG is rated as Early (TRL 1-3).

2.7.3.6.4 (U) Standards

(U) Table 2.7-8 lists the RFID standards applicable to Inventory Management

Table 2.7-8: (U) Inventory Management RFID Standards

This Table is (U)

Name	Description
EPC Global Network Standards	
EPC Tag Data Specification Version 1.1	Identifies the specific encoding schemes for a serialized version of the EAN.UCC Global Trade Item Number (GTIN®), the EAN.UCC Serial Shipping Container Code (SSCC®), the EAN.UCC Global Location Number (GLN®), the EAN.UCC Global Returnable Asset Identifier (GRAI®), the EAN.UCC Global Individual Asset Identifier (GIAI®), and a General Identifier (GID)

This Table is (U)	
Name	**Description**
900 MHz Class 0 Radio Frequency (RF) Identification Tag Specification.	This document specifies the communications interface and protocol for 900 MHz Class 0 operation. It includes the RF and tag requirements and provides operational algorithms to enable communications in this band.
13.56 MHz ISM Band Class 1 Radio Frequency (RF) Identification Tag Interface Specification.	This specification defines the communications interface and protocol for 13.56 MHz Class 1 operation. It also includes the RF and tag requirements to enable communications in this band.
860MHz -- 930 MHz Class 1 Radio Frequency (RF) Identification Tag Radio Frequency & Logical Communication Interface Specification	This document specifies the communications interface and protocol for 860 – 930 MHz Class 1 operation. It includes the RF and tag requirements to enable communications in this band.
Physical Markup Language (PML)	The PML Core specification establishes a common vocabulary set to be used within the EPC global Network. It provides a standardized format for data captured by readers. This specification also includes XML Schema and Instance files for your reference.
ISO Standards	
ISO/IEC 15963:2004	Information technology -- Radio frequency identification for item management -- Unique identification for RF tags
ISO/IEC 18000-4:2004	Information technology -- Radio frequency identification for item management -- Part 4: Parameters for air interface communications at 2.45 GHz
ISO/IEC 18000-6:2004	Information technology -- Radio frequency identification for item management -- Part 6: Parameters for air interface communications at 860 MHz to 960 MHz
ISO/IEC 18000-7:2004	Information technology -- Radio frequency identification for item management -- Part 7: Parameters for active air interface communications at 433 MHz
This Table is (U)	

2.7.3.6.5 (U) Cost/Limitations

(U) Tags vary significantly in cost, depending upon their frequency range, application, and whether they are active, semi-active, or passive. Current industry efforts are working to reach the goal of $0.05 for a passive smart-label tag, at which point a host of applications become economically feasible. Current tags range from $100 for complex, long-range active tags, to approximately $.50 to $1.00 per tag in very high-volume applications. The major limitation for GIG IA applications will be the cost of tags which can support the encryption and authentication required to securely deactivate and reactivate RFID tags.

(U) Readers vary in cost depending upon the type and range requirements. Fixed installation systems with separate antennas can cost several thousand dollars. RFID readers in a PC Card (PCMCIA) format are currently available for $150.

2.7.3.6.6 (U) Alternatives

(U) Standard optical bar codes are an alternative to RFID tags, but they carry serious limitations. Bar codes require line of sight to read so they must be external to packaging, unobstructed, and facing the reader. This may require manual orientation of the scanner or the scanned item. Bar codes can only be read one at a time by a scanner. Because they are exposed, printed barcodes are susceptible to wear, dirt, marks, and water, or chemical damage, becoming unreadable. In contrast, RFID tags can be sealed inside a relatively impervious container.

2.7.3.6.7 (U) Complementary Techniques

(U) RFID tagging systems only provide value when tied to updates of a centralized, real-time asset management application. The application provides visibility into the inventory status, and the RFID system provides real-time, highly accurate updates to the inventory.

2.7.3.6.8 (U) References

(U) [Chung] "Low Cost and Reliable RFID Tags for All Frequencies," by Kevin Chung, http:// itpapers.zdnet.com/ abstract.aspx?kw=%20RFID&dtid=1&docid=89816

(U) [DOD04] "Radio Frequency Identification (RFID) Policy," Undersecretary of Defense for Acquisition, Technology and Logistics (USD/ATL) Memorandum, July 30, 2004. http://www.acq.osd.mil/log/logistics_materiel_readiness/organizations/sci/rfid/assetts/Policy/RFID%20Policy%2007-30-2004.pdf

(U) [Finkenzeller03] "Frequencies for RFID-systems," by Klaus Finkenzeller, from RFID Handbook, 2ed, tr. Rachel Waddington, Wiley & Sons, Ltd, April 2003.

(U) [Hodges03] "Demystifying RFID: Principles and Practicalities," by Steve Hodges, Mark Harrison, October 1, 2003. www.autoidlabs.org/whitepapers/CAM-AUTOID-WH024.pdf

(U) [Juels03a] "Minimalist Cryptography for Low-Cost RFID Tags," by Ari Juels, 2003. http://www.eicar.org/.../ 11%20-%20Mynimalist%20Cryptography%20for%20RFID%20Tags.pdf

(U) [Juels03b] "The Blocker Tag: Selective Blocking of RFID Tags for Consumer Privacy," by Ari Juels, Ronald L. Rivest, Michael Szydlo. In V. Atluri, ed. 8th ACM Conference on Computer and Communications Security, pp. 103-111. ACM Press. 2003. http://www.rsasecurity.com/rsalabs/node.asp?id=2060

(U) [Ohkubo03] "Cryptographic Approach to "Privacy-Friendly" Tags," by Miyako Ohkubo, Koutarou Suzuki, Shingo Kinoshita, November 2003.
(U) http://www.rfidprivacy.org/papers/ohkubo.pdf

(U) [Rivas03] "RFID – its Applications and Benefits," Mario Rivas, RFID Privacy Workshop @ MIT: November 15, 2003. http://www.rfidprivacy.org/papers/rivas/rivas.pdf

(U) [Sarma03] "RFID Systems and Security and Privacy Implications," by Sanjay E. Sarma, Stephen A. Weis, Daniel W. Engel, http://www.eicar.org/rfid/kickoffcd/04%20-%20Hintergrundinformationen/09%20-%20RFID%20Systems%20and%20Security%20and%20Privacy%20Implications.pdf

2.7.3.7 (U) Compromise Management of IA Devices

2.7.3.7.1 (U) Technical Detail

(U//FOUO) Compromise Management is the management and actions required to respond to the potential compromise of IA Devices. A device is compromised when the integrity and confidentiality of data on that device cannot be assured or determined. Many IA Devices will be operating in unprotected, partially protected or tactical environments where they may fall into the hands of an adversary. At that point the capability to use the equipment to communicate on the GIG must be removed.

(U) Compromise Management consists of the following components:

- (U) Compromise Detection
- (U) Compromise Investigation
- (U) Compromise Isolation
- (U) Compromise Recovery.

(U//FOUO) Compromise Detection is the ability to determine that an IA component has been tampered with, either physically or logically. Many components have mechanisms to indicate when tampering has occurred. Mechanisms that may indicate the physical integrity of a component include:

- (U) Physical labels that tear easily
- (U) Tamper detection hardware, included in the component as part of the design
- (U) Audit logs or alarms also form a component of compromise detection. These are discussed further in Section 2.7.3.8, Audit Management
- (U) Explicit regular external communication to check the status of the component. This may be in the form of a SNMP status check or keep-alive timers on a physical link
- (U) In the GIG environment, IA devices will spend more and more time in less and less protected environments, and security will be dependent upon the internal IA device protection or the network's ability to detect device or system compromise.

(U) Compromise Detection – Tamper Mechanisms. The first key technology supporting compromise detection is tamper resistance and detection. Tamper resistance is the use of physical packaging to restrict the ability to physically alter or connect to components of a device. Tamper detection is the addition of elements to the component to provide an active indication to the system that a compromise is taking place. In many situations today, tamper detection is done through physical means, such as seals. Seals can be applied to any physical enclosure or opening to determine if an attempt has been made to open it. However, such mechanisms require physical inspection by a knowledgeable person to determine if tampering may have occurred. Instead, active measures must be incorporated into IA components to detect attempts to tamper with them or compromise their integrity.

(U) All high-assurance cryptographic modules must provide a means to detect tampering. NIST FIPS 140-2 specifies federal requirements for cryptographic modules. For security level 2 modules, they must provide coatings or seals that will make tampering evident. It specifies that for security level 3, components must zeroize any keys or sensitive parameters whenever the device is opened. For level 4, components must have a high probability that any attempt to tamper with the device or bypass the physical protection will result in device zeroization. A wide variety of techniques are used to detect tampering, such as:

- (U) Switches on access panels or lids
- (U) Temperature sensors to detect attempts to manipulate the device by operating it outside normal temperature parameters
- (U) X-ray sensors to detect attempts to image the interior circuitry
- (U) Ion-beam sensors to detect attempts to probe specific integrated circuit gates
- (U) Voltage sensors to detect attempts to operate the device outside its normal voltage parameters to force lockups or processing vulnerabilities
- (U) Wire or optical fiber meshes assembled over components and sealed inside sealing compounds that are wired to detect holes 50 um or larger.

(U) In high-assurance components, a permanent battery powers these sensors for the life cycle of the component, so that they are active even when the device is powered down or being shipped. The standard response is that any keys or security parameters are zeroized or cleared. Due to issues with standard static RAM remnants, this operation is considerably more complex than simply removing power to SRAM memory. It generally involves at least writing multiple times to each location to overwrite data.

(U) Compromise Detection – Keep Alive Protocol. The current technology for external keep-alive testing is the SNMP. Currently this is widely implemented as part of network management products and is used for network status reporting, covered at length in Section 2.6.

(U) Compromise Investigation is the ability to determine with a high assurance that a component is either operating within its parameters or that it cannot be determined. Since many compromise detection approaches are indirect, and only provide evidence of tampering, further investigation may be required. This is a verification of the configuration of an IA device. This is described in Section 2.7.2.5.

(U//FOUO) Compromise Isolation is the ability to isolate a component that is no longer trusted from the rest of the GIG. There are two components of this. The first is the reliable removal of any keys from the IA component, or zeroization. The second is the notification of all other GIG entities that may communicate with or use a component that it is not trustworthy. This is accomplished through such mechanisms as CRLs or the Online Certificate Status Protocol (OCSP). This is described in Section 2.7.2.4. For IA devices that do not use the PKI infrastructure, key replacement is described in Section 2.7.2.3.

(U) Compromise Recovery is the ability to restore a device to operation after its integrity has been restored. In many cases, the compromise of a device may be temporary or in error—in which case the device must be restored to service. There are two facets of Compromise recovery. First, the configuration of the component must be restored. This means the software, data, and firmware must be restored to a known, assured state, either by verification of the existing configuration of the component or by reinitializing it and restoring the configuration. This is described in Section 2.7.2.5, Configuration Management. Second, the trustworthiness of the device must be communicated to its peers. These are certificate and key management issues, which are discussed in Sections 2.7.2.4, Certificate Management and 2.7.2.3, Key Management, respectively.

2.7.3.7.2 (U) Usage Considerations

2.7.3.7.2.1 (U) Advantages

(U) These mechanisms are required for high-assurance devices such as INEs or HAIPEs that protect Secret or above data. FIPS 140-2 requires them for Level 4 devices used for high-assurance unclassified operations.

2.7.3.7.2.2 (U) Risks/Threats/Attacks

(U/FOUO) The number and types of possible physical tampering attacks against IA devices number in the hundreds [Weingart00]. We describe some of the broad characteristics of attacks that must be considered.

(U) Physical threats to IA enabled equipment have been characterized by three classes of attackers:

- (U) Class 1 - clever outsiders – It is assumed the attacker has limited knowledge of the system, but can take advantage of known weaknesses. This typically characterizes hackers.
- (U) Class II -knowledgeable insiders – They have substantial specialized technical experience and highly sophisticated tools and instruments. They include professional researchers and academics.
- (U) Class III funded organizations – Specialists backed by large funding sources, capable of in-depth analysis, sophisticated attacks, and extremely advanced analysis tools. These include criminal organizations and foreign governments.

(U) The attacks can be characterized as well by the goal of the attacker:

- (U) Steal keys – The attacker wants to extract unencrypted keys or cryptographic parameters protected by a device for loading into another device
- (U) Use equipment to continue communication – The attacker wants to control the device and use it to continue communications for intelligence or further attacks
- (U) Reverse engineering – The attacker wants to copy the device

- (U) Backdoor the device – The attacker wants to modify the device with a backdoor or Trojan without detection and allow its continued use while stealing data or further compromising the network.

(U) Each of these goals affects the type of attack from relatively simple non-invasive, non-destructive, attacks to invasive attacks which modify or destroy the device under attack.

2.7.3.7.3 (U) Maturity

(U) The mechanisms of tamper detection are understood, and current commercial products are available that incorporate them. However, in many cases the tamper response is limited to zeroizing the sensitive contents of the IA device. Currently only a few type 1 cryptographic devices, (e.g., HAIPE-compliant products) support SNMP management and so are physically capable of network alerts of tampering. However, current security policy is that tamper detection results in an immediate, non-interruptible response of zeroizing all communications keys, making it impossible for a device to securely send any communications such as a tamper indication to a central manager. Most commercial cryptographic modules only incorporate passive tamper resistance, only one device, the IBM 4578 cryptographic processor was evaluated to FIPS 140-1 Level 4 which mandates tamper detection. The Dallas Semiconductor DS5240 and DS5250 processors incorporate tamper detection but have not been FIPS evaluated. SNMP management of network devices is standard, and as additional commercial implementations of the specification emerge, network notification of tamper will become commercially available.

(U) The maturity of compromise management technology is assessed as Emerging (TRLs 4 - 6). Commercial products with limited capabilities are available. However, they are expensive and are not widely used or supported. Current GOTS equipment routinely incorporates zeroizing as a compromise response, but current designs do not define any possible mechanism by which communications with a management entity can occur after a zeroization. External compromise detection by keep-alive or heartbeat protocols can be implemented by current standard protocols, but no provision for explicit compromise signaling or detection exists.

2.7.3.7.4 (U) Standards

Table 2.7-9: (U) Compromise Management Standards

This Table is (U)	
Name	**Description**
NIST Standards	
FIPS 140-2	Security Requirements for Cryptographic Modules
IETF Standards	
SNMPv3	The Simple Network Management Protocol, version 3 is the latest version of the IETF standard for managing network devices. Version 3 includes authentication and authorization, so it is considered much more secure than previous versions. SNMP is widely implemented, but has some significant restrictions because of its very simple structure.
ISO Standards	
ISO/IEC 15408-1:1999	Information technology -- Security techniques -- Evaluation criteria for IT security -- Part 1: Introduction and general model

This Table is (U)	
Name	**Description**
ISO/IEC 15408-2:1999	Information technology -- Security techniques -- Evaluation criteria for IT security -- Part 2: Security functional requirements
ISO/IEC 15408-3:1999	Information technology -- Security techniques -- Evaluation criteria for IT security -- Part 3: Security assurance requirements
This Table is (U)	

2.7.3.7.5 (U) Cost/Limitations

(U) Cost is the major limitation on the use of tamper mechanisms. As a result, tamper mechanisms are only implemented on high-assurance cryptographic equipment, either FIPS 140-2 Level 4 or Common Criteria EAL 4, or above. The manufacturing complexity and limited production of such components has meant that components incorporating tamper mechanisms are extremely expensive relative to components certified to lower assurance levels.

2.7.3.7.6 (U) Complementary Techniques

(U) The primary complement to tamper mechanisms is an external approach using a keep-alive protocol between the IA Component and an external source such as the Network Operations Center. Common protocols such as ICMP were designed for testing a connection or the response from a server. However continuing issues with using ICMP for DoS attacks has meant that it is often turned off and certainly restricted to within an enclave.

(U) The TCP includes the notion of a keep-alive packet that essentially checks at regular intervals to see if the connection has been dropped on an otherwise idle TCP connection. It is a null packet that serves only to generate a TCP disconnect if it does not go through. The negative is that it only indicates that the connection failed, which can be due to transient network conditions, and does not reflect the state of the connection endpoint host. However, a TCP connection does consume network resources on both ends, so it does not scale well to large numbers of systems.

2.7.3.7.7 (U) References

(U) [Anderson96] "Tamper Resistance – a Cautionary Note, in Second USENIX Workshop on Electronic Commerce Proceedings," by Ross Anderson, Markus Kuhn; Oakland, CA. 1996. http://www.cl.cam.ac.uk/users/rja14/tamper.html

(U) [ATMEL04] "AT97SC3201 The Atmel Trusted Platform Module," Atmel Corporation, www.atmel.com/dyn/resources/prod_documents/doc5010.pdf

(U) [Auer00] "Tamper Resistant Smartcards – Attacks and Countermeasures," by Auer, Eric; http://www-krypt.cs.uni-sb.de/teaching/seminars/ss2000/auer.pdf

(U) [Bajikar02] "Trusted Platform Module (TPM) based Security on Notebook PCs – White Paper," by Sundeep Bajikar, June 20, 2002. developer.intel.com/design/mobile/platform/downloads/Trusted_Platform_Module_White_Paper.pdf

(U) [CSVP04] FIPS 140-1 and FIPS 140-2 Vendor List, NIST Cryptographic Standards and Validation Program, 2004. http://csrc.nist.gov/cryptval/140-1/1401val-all.htm

(U) [Dallas03] “DS5250 High-Speed Secure Microcontroller,” Dallas Semiconductor, July 18, 2003. http://pdfserv.maxim-ic.com/en/ds/DS5250-DS5250F.pdf

(U) [Johnston97] "Vulnerability Assessment of Security Seals," by R. G. Johnston and A. R. E. Garcia, Journal of Security Administration, 20, 15 (1997, http://lib-www.lanl.gov/la-pubs/00418796.pdf.

(U) [Weingart99] “The IBM 4758 Secure Cryptographic Coprocessor Hardware Architecture and Physical Security,” by S. H. Weingart, IBM Corporation, 1999. http://www.cl.cam.ac.uk/Research/Security/seminars/1999/materials/weingart-19990222b.pdf

(U) [Weingart00] “Physical Security Devices for Computer Subsystems: A Survey of Attacks and Defenses,” S. H. Weingart, Workshop on Cryptographic Hardware and Embedded Systems, 2000.

2.7.3.8 (U) Audit Management

2.7.3.8.1 (U) Technical Detail

2.7.3.8.1.1 (U) Audit Life Cycle

(U) The typical lifecycle of an Audit process can be seen in Figure 2.7-11.

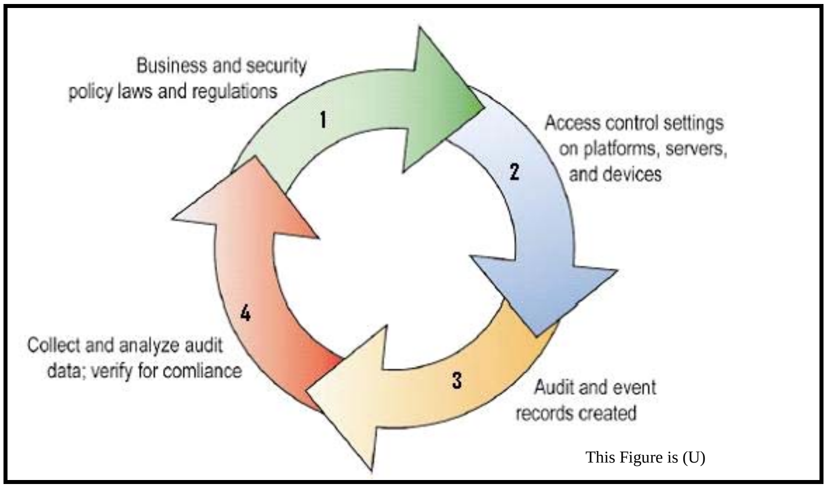


Figure 2.7-11: (U) Audit Life Cycle

(U) Business and security policies form the first step in an audit life cycle. Policies are then implemented via access controls that are put in place to enforce the rules of the policies. Access controls mandate how users are authenticated and granted access to system resources. As users conduct their business functions, Identity Management and other system components generate audit events that are stored locally in log files or forwarded to event log databases. Finally, audit and event data is collected and analyzed to verify that the intent of the business and its security policies has been carried out.

2.7.3.8.1.2 (U) Auditing – Objectives

(U) Policy Compliance: Enterprises, such as the GIG, use systems and services that will need to comply with business, security, legal, and regulatory mandates, such as SOX (Sarbanes-Oxley), HIPAA, FISMA, DCID 6/3, and NISPOM. Thus, audit and event records need to be recorded and monitored in order to provide the evidence that the GIG will use to demonstrate compliance.

(U) Detecting Intrusions: Auditing is the ability to provide the means of detecting events that result in a security breach of the GIG system. As such, the audit management of event logs works closely with the collection services of the IDS and IPS systems. It is the latter’s objective to collect, analyze, detect, and react to the event log data for intrusions.

(U) Determining Performance: Auditing also provides a means of independent review and examination of records to determine the adequacy of system controls that ensure compliance with established policies and operational procedures. This information serves as a resource for the recommendation of necessary changes in controls, policies, and procedures. Auditing of system resources should provide the information needed to reconfigure these resources to improve system performance.

(U) Accountability: Auditing will be used to identify an individual, process, or event associated with any security-violating event. In order to provide a complete audit picture, data must be collected and classified according to one of a number of areas of concern. This multi-dimensional approach would include an audit recording based on a subject's attributes, a time tagged object, and the state of a system resource. The subject will be tied to an audit event record via the individual's identification data, if that is tagged appropriately.

(U) Access to the GIG will require authentication of the individual attempting to log into the system. The user login event will be recorded in the audit log along with any security-related audit events associated with the individual user. An identifier that will uniquely identify the user will be logged for these events. Object-based auditing identifies an audit event by an identifier of a modifiable security related data item such as a file on a storage medium. This identifier must include the name of the file and the storage volume identifier. An audit event would be generated whenever a security related object, such as a configuration file, was modified. The resource identifier is used in the auditing of system resources, such as network throughputs or the percentage of idle time during specified intervals or periods.

(U) Robustness: Audit logs and the data contained in them represent valuable information, especially to adversaries who are attempting without detection to intrude and compromise a system. Such undetected activities of intruders could wreak significant havoc, such as the unleashing of malware, denial of service attacks, espionage, and other harm. Consequently, audit data and services must be strongly secured, employing the most robust access control standards possible for each situation.

(U) Log Analysis: Logs and event records created by infrastructure systems are part of the evidence trail of what happens during the course of business for an enterprise. By examining audit logs, GIG systems can determine whether security components are properly enforcing policies and regulations to provide accountability in the event that non-compliance occurs. Audit log analysis can also reveal valuable information on patterns and exceptions. Long-term trends or usage patterns can help system planners adjust to customer habits; support forensic analysis for investigations into fraudulent activity; and harden targeted servers on sensitive systems that are experiencing attacks. Monitoring audit and event data in real time can enable enterprises to react to attacks in progress or new threats as they emerge.

2.7.3.8.1.3 (U) Audit Trails – Flow, Formats and Storage

(U) Figure 2.7-12 shows the typical information flows associated with audit trails. Audit records are generated at sources that include network devices, operating systems, and applications. The records are transported either internally within systems using inter-process communications or over networks using network protocols to storage media. Stored audit information along with other information from the system is either analyzed within the system under examination or by using separate analysis stations.

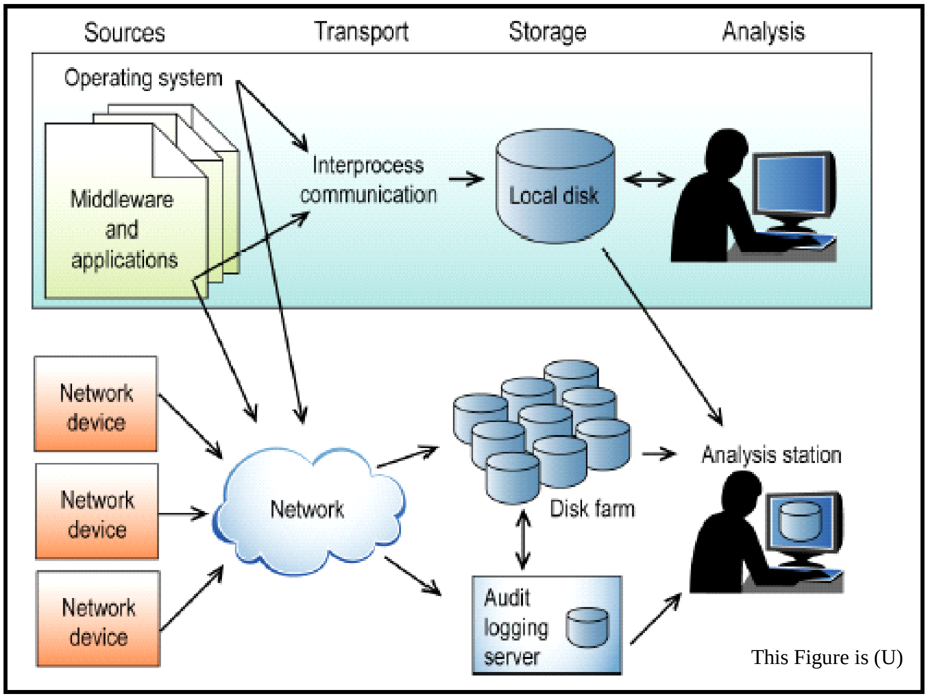


Figure 2.7-12: (U) Audit Trail Information Flow

(U) Audit sources generate audit records in a wide variety of formats and transmit and store them using a range of different techniques. Communication within a single system via inter-process communication is usually effective at retaining integrity, but storage within the system under examination makes these records subject to attack by anyone circumventing system security. Analysis of audit trails and the systems they are supposed to reflect can be quite complex and time consuming depending on the audit's objectives. Analysis within the system under examination creates audit integrity and other related problems. With the exception of low assurance casual audits, audits within trusted systems, and special cases where there are no other options, information flows that remain entirely within a single system should be avoided.

2.7.3.8.1.4 (U) Providing Reports.

(U) An important aspect of Audit management is the ability to provide Conformance and Compliance Reports to show that user and system activities are indeed complying with the governing policies. These compliance reports should be generated automatically, periodically, and on demand.

(U) Compliance reports are used by auditors and review management. The reporting technology should provide many types of views to help management visualize the findings and take appropriate action based on the assessments. Higher levels of reviewing typically involve Visualization and UI (User Interface) reporting tools that visually depict details or summaries in multiple dimensions (3D), indicate weak points or failures, provide overviews of the operational security health of the infrastructure, as well as indicate conformance to policy, compliance, or lack thereof. These reports can be useful in conducting further risk analysis, as well as for improving the process and resource provisioning of the system.

2.7.3.8.2 (U) Usage Considerations

2.7.3.8.2.1 (U) Implementation Issues

2.7.3.8.2.1.1 (U) Monitoring and Verification of Compliance to Policy

(U) A number of policy categories are required to be supported:

- (U) Regulatory policies: FISMA, DCID 6/3, NISPOM, SOX (Sarbanes-Oxley), etc.
- (U) Intrusion Detection (IDS, IPS) based policies
- (U) Configuration Management policies, such as software and hardware upgrade policies. These include IAC CM policies, such as the updates and patches applied to application software, virus detection software, etc.

(U) There is technology currently available that aids in capturing and applying policy statements via software tools and then using the stipulated policy rules to monitor and verify that the system or enclave activities are taking place within the rules. However, the main issue of concern is that today's solutions are mainly point solutions. Each vendor's software is proprietary in nature, differs from the others, and as such best of breed components from among different vendor choices cannot be selectively mixed. The major reason for this is the lack of standards that would allow policy rules to be specified and monitored in a uniform and normalized manner. Hence, there is little interoperability between vendor products.

2.7.3.8.2.1.2 (U) Tamper Resistance of Logs

(U) The following assertions and discussion are based on Figure 2.7-13:

(U) Low Assurance Architectures are Usually Inadequate:

(U) A low assurance architecture has an auditor logged into the system while it is operating. This presents the potential for the auditor to alter and affect the system, for the auditor to be fooled by the system under examination, for those under audit to detect the presence of the auditor, for the auditor to damage the system under examination, for audit trail loss or damage, or for the revelation of audit records in unauthorized ways. Such audit architectures should only be used in low-risk situations (low threat and low consequence), involving audits that are not related to regulatory compliance and where any of these consequences from the audit are acceptable.

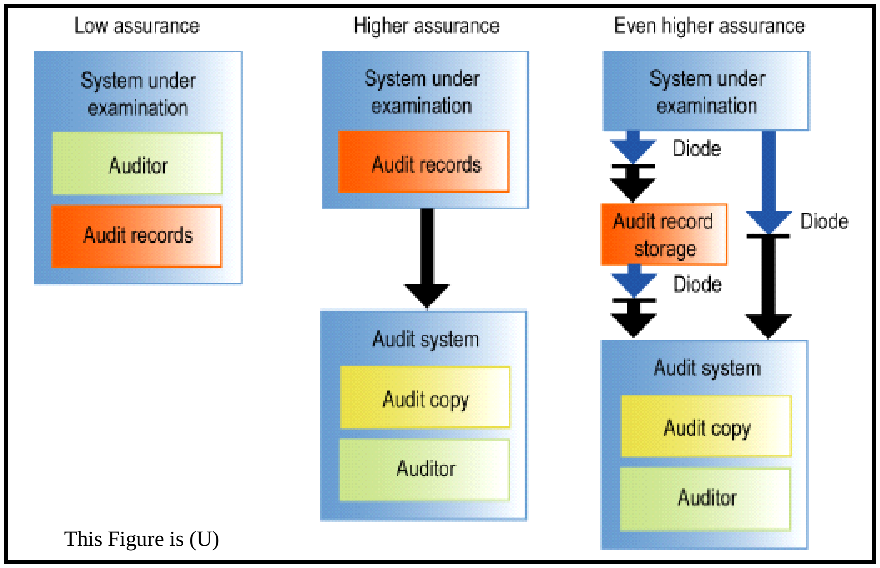


Figure 2.7-13: (U) Audit Logs – Protection

(U) Higher Assurance Architectures are Advised for Medium Risks:

(U) A higher assurance architecture separates the auditor from the system under examination. If properly implemented, no information flows from the audit system to the system under examination, and the audit system includes a copy of all audit records as well as a forensically sound copy of the contents of the system under examination. In this example, audit records can be attacked within the system under examination, but the auditor can have no effect on that system or the audit records. It is impossible for the users of the system to know from the system whether an audit is underway, and the auditor can operate without concern about harm to the system under examination or subversion by the system under examination. This architecture is acceptable in most medium-risk situations (medium or lower threats and medium or lower consequences) and is normally acceptable for regulatory compliance audits in cases when loss or subversion of audit records is acceptable. In cases where audit records are required, such as under Gramm Leach Bliley regulations, this approach is inadequate, because the original audit records can be subverted.

(U) Even Higher Assurance Architectures are Advised for High Risk:

(U) An even higher assurance architecture adds independent audit trail storage and higher assurance separation of the audit trails and auditor from the system under examination. The use of digital diodes (systems that enforce one-directional information flows) provides high assurance against backflows of information, while the use of an external audit record storage device separates the audit records from those who might seek to subvert the audit trail. The auditor is protected against subversion, the system is protected from the auditor, and the audit records are protected from attackers. For additional assurance, redundant copies of audit trails can be generated and stored, additional coding can be used to verify records in transmission and storage, records can be generated from multiple sources associated with the system under examination, and higher assurance components can be used. There are audit servers on the market designed to implement the audit record storage requirements of this architecture, and most system audit mechanisms provide the means to transmit audit records as they are generated to remote systems over a network. Some audit servers also provide reasonable assurance against information backflows, forming different assurance levels of diode protection. This network audit architecture should be used for situations in which threats or consequences are high and regulatory compliance mandates effective auditing.

2.7.3.8.2.1.3 (U) Log Formats and Event Records

(U) A lack of standard message formats and exchange protocols intensifies the problem of coping with the huge data volume. Operating systems, firewalls, application servers, intrusion detection systems, and other network components create proprietary record formats that must be normalized before additional correlation analysis can be performed. Auditing systems, including directory, access management, and provisioning servers, contribute to the chaos with their mostly inadequate auditing features that require manual handling of nonstandard records, and often with no unified audit view within their product boundary—and certainly none beyond it. Without standard exchange protocols, software vendors have little reason to do more than write their own proprietary log records, and audit tools vendors are forced to write platform-specific agents, parse diverse log file formats, or rely on sparse protocols like the SNMP to transmit data and event information to central servers.

(U) A partial list of supported devices, each with their own vendor-specific log formats:

- (U) Firewall products
- (U) Antivirus products
- (U) Intrusion detection products
- (U) Routers and switches
- (U) SYSLOG

(U) Various devices that record and log events are shown in Figure 2.7-14.

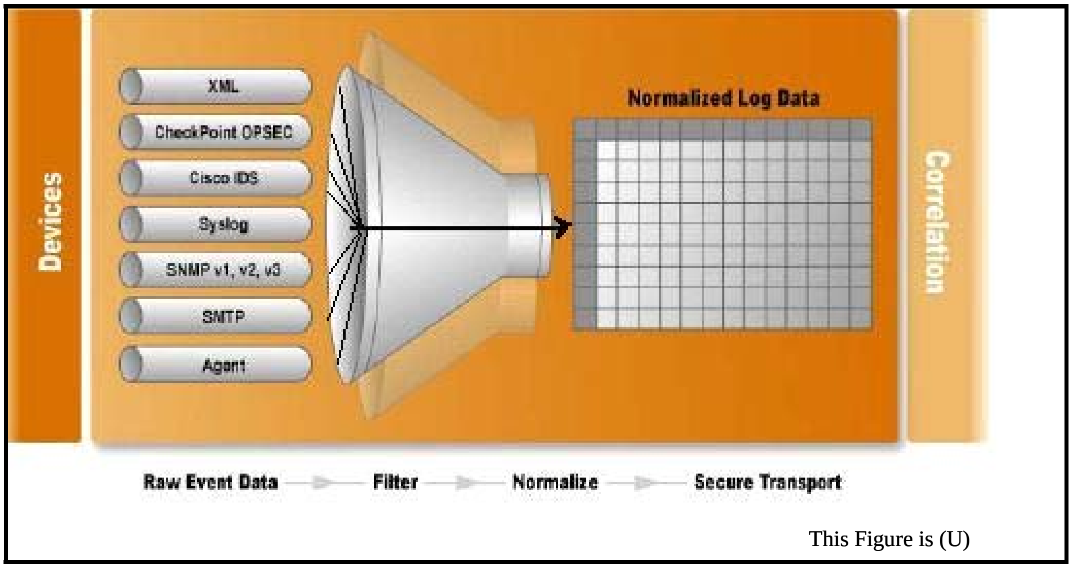


Figure 2.7-14: (U) Aggregation and Normalization

(U) Audit log data, indicated as raw event data in Figure 2.7-14, is collected from various devices and sensors on the network. This raw data are either typically pulled or monitored from a central monitoring facility, or are pushed out the devices via agent technologies. Regardless of the manner in which the data is collected, the data then undergoes filtration, aggregation, and normalization into a unified format before it is further transported (securely) to analytical and correlation engines for intrusion or anomaly detection. Current technologies for normalization are manifested in the form of custom middleware that performs the normalization into formats only understood by the custom vendor provider. This is because standards that provide normalized formats do not currently exist. As such, normalization is subject to vendor interpretation and consequential errors.

(U) The type of attributes surrounding auditable events also vary among the various devices, sensors, operating systems, and platforms. Due to lack of standards or policies, not all implementations of log events capture the following essential attributes:

- (U) Subject – The person accessing the object
- (U) Object – The target object that is accessed by the subject
- (U) Resource – Monitor items like throughput and idling time, used for performance and utilization measurements
- (U) Time Stamps – When the activities occurred
- (U) Event Status – Success/Failure with appropriate codes.

2.7.3.8.2.1.4 (U) Collection Services

(U) Push versus Pull Agents: Some software vendors provide agents to forward logs to a Security Operations Center (SOC); this is considered a push model, since the host or target device pushes data out to the collection side via a custom host agent. There are also central monitoring services that are agent-less; that is, they do not require forwarding agents at host sites, but instead use technology to pull device logs from a central facility (e.g., SOC). These central monitoring services poll the distributed network of hosts and remote devices' logs at specified intervals. These collection modes can be seen in Figure 2.7-15.

(U) There are currently no standards-based specifications that prescribe interfaces between central and host agents—or specify how to achieve interoperability.

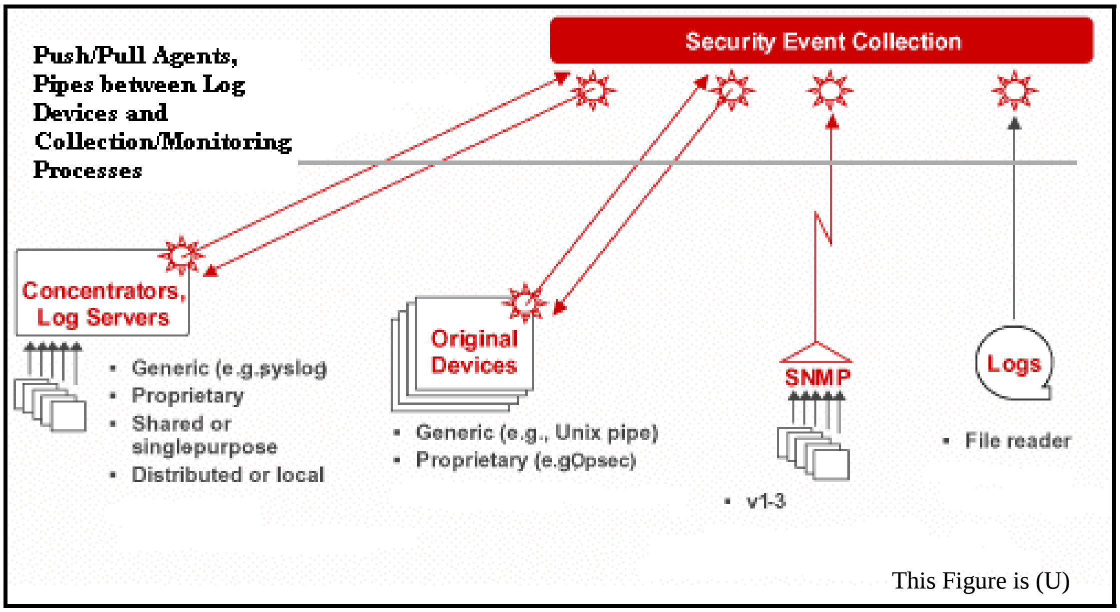


Figure 2.7-15: (U) Interfaces - Agents and Pipes between Log Devices and the Collection/Monitoring Processes

2.7.3.8.2.1.5 (U) Log Reduction and Archiving (Log Retention)

(U) Event reduction and full logging are opposing methodologies. The motivation for event reduction is to reduce the event data set in order to quickly detect deviations from the operational norm for IDS and IPS purposes; this is achieved by filtering out the noise or non-threat event information. Whereas, full logging and complete archives capture and log every event. This is done for computer forensics needs that include criminal investigations, where the complete audit trail archive is a requirement for the chain of evidence. The extent of log archives, reduction and retention will depend on the situational system policy or policies being established. Policies that require both Event Reduction (e.g., for IDS and IPS needs) and full logging (Forensic needs) need to be supported.

2.7.3.8.2.2 (U) Advantages

(U) Management can be implemented with or without agents.

(U) Agent technologies are more suited for host-based logging and monitoring. They have the advantage of being able to log events even when the connection to the network goes down or is unavailable for any reason. Agent technologies tend to push data out of the logs, periodically, to a central monitoring service on the network. They do require periodic configuration and maintenance however.

(U) Agentless technologies are built into centralized monitors and have the distinct advantage of eliminating the need for host-based agents. This eliminates the maintenance that would otherwise be required on a host system. But the central monitors do have a few disadvantages. They depend on the availability of the network. Central monitoring is also more complex; it requires keeping an up-to-date list of target hosts and routers whose logs need monitoring.

(U) Management can also include all logs, or reduced logs.

(U) Managing Full-logs (i.e., no reduction) has the advantage of being simpler to implement. But this also imposes higher stress levels on network bandwidth and storage.

(U) Log reduction management is just the opposite. It works well with comparatively modest bandwidth and storage requirements, but requires the maintenance of complex analytical software that can accurately filter out non-threat noise from the real threat related events.

2.7.3.8.2.3 (U) Risks/Threats/Attacks

(U) Audit logs run the risk of being a target for attack due to the valuable information contained in them. Thus, managing the audit data requires high assurance and tamper resistance. Assurance implies the confidentiality, integrity and continuous availability of the audit trails and logs data.

(U) Audit data represents valuable policing information and is thus highly desirable as a target for attack, stealing, or modification. Consequently audit data should be protected from unauthorized access or compromise and needs to be secured at every step whether the audit data is at rest in a latent log file or in motion (transported over the network for analysis and post processing). Appropriate access-controls and hardening principles need to be in place to ensure the integrity and proper authorized access to the audit event data.

(U) Audit data should also be made available, on demand, for urgent or immediate processing needs. Thus, provisions for continuous availability of the data are required for consideration. This would include backup and fault tolerant audit databases.

(U) Audit technologies are also affected by the dynamic nature of policy changes. Dynamic Policy Management states that policies and their rules can change dynamically based on situational and directive changes. Consequently, auditing mechanisms are then at the risk of being outdated quickly, and if the technologies do not permit the adaptability of auditing processes to new policies and rules, then false positive or false negative reporting can occur as a result—thereby defeating the auditing mission.

2.7.3.8.3 (U) Maturity

(U) The Audit management market appears to be somewhat mature today, but products exist only as point solutions. As indicated earlier, vendor solutions can be found in the SEM market today. The SEM vendors provide all the middleware that tie together the various steps of audit monitoring, collection, filtering, and normalization. But their solutions are proprietary in nature, and there is little or no interoperability between the various facets of secure event management and auditing capabilities.

(U) The efforts of groups like IDMEF, CIDF, and CERIAS are still largely unknown. They have yet to emerge with concrete standards and are outlined in next section on Standards.

(U) Audit Management today exhibits a lack of maturity in standards-based solutions. This makes componentization and interoperability in the different phases of audit management very difficult. Standards are needed to prescribe log formats, normalized records, interfaces with collection processes, and policies directed towards secure storage as well as secure transport mechanisms to and from hosts and collection/analytical agencies.

(U) Audit management technologies are assigned an overall maturity level of Emerging (TRLs 4 - 6). This is based on the middleware technologies (point solutions) available in the commercial SEM market. However, standards for GIG-wide audit log formats, aggregation and normalization of records, and interfacing to audit analysis processes that include IDS and IPS systems, need to be devised and adopted.

2.7.3.8.4 (U) Standards

(U) A general lack of standards is one of the main challenges to the collection and correlation of security events from heterogeneous systems. The few existing standards (Table 2.7-10) are still in development, have not gained significant acceptance in the industry, or are narrowly focused on a particular technology area. Some vendors have started using eXtensible Markup Language (XML) to describe the event records in their repositories, but the formats are still proprietary.

Table 2.7-10: (U) Audit Management Standards

This Table is (U)	
Name	**Description**
IETF Standards	
CLF	Common Log Format. Typically, the information is presented in plain ASCII without special delimiters to separate the different fields. See http://www.ietf.org
ELF	Extended Log Format
IDMEF	Intrusion Detection Message Exchange Format. The IETF's Intrusion Detection Working Group (IDWG) is developing message formats and procedures for sharing messages between intrusion detection systems and the SEM systems that manage them. The IDMEF requirements were posted as an Internet Draft in October, 2002, along with a draft of the Intrusion Detection Exchange Protocol (IDXP). In January, 2003, an Internet Draft was submitted for IDMEF that included an XML implementation. This initiative is still in development and it’s future is uncertain.
RFC 1155,	Structure of Management Information

This Table is (U)

Name	Description
RFC 1156	Management Information Base (MIB-I)
RFC 1157	SNMP
RFC 1187	Bulk table retrieval
RFC 1212	Concise MIB definitions
RFC 1213	Management Information Base (MIB-II)
RFC 1215	Traps
RFC 1227	SNMP Multiplex (SMUX)
RFC 1228	SNMP-DPI
RFC 1229	Generic-interface MIB extensions
RFC 1239	Reassignment of MIBs
RFC 1243	AppleTalk MIB
RFC 1248	OSPF MIB
IEEE Standards	
1230 IEEE 802.4	Token Bus MIB
1231 IEEE 802.5	Token Ring MIB
ISO Standards	
ISO 8824-1	Abstract Syntax Notation One (ASN.1): Specification of basic notation
ISO 8824-2	Abstract Syntax Notation One (ASN.1): Information object specification
ISO 8824-3	Abstract Syntax Notation One (ASN.1): Constraint specification
ISO 8824-4	Abstract Syntax Notation One (ASN.1): Parameterization of ASN.1 specifications
ISO 8825-1	ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)
ISO 8825-4	ASN.1 encoding rules: XML Encoding Rules (XER)
Other Standards Efforts	
Common Intrusion Detection Framework	http://gost.isi.edu/cidf/ Taken from above website: "The Common Intrusion Detection Framework (CIDF) is an effort funded by DARPA to develop protocols and application programming interfaces so that intrusion detection research projects can share information and resources and so that intrusion detection components can be reused in other systems." It appears that the CIDF initiative started in 1997, but has yet to materialize into an accepted standard. The work is still under development.
Open Security Exchange	The Open Security Exchange in April, 2003, announced specifications to enable more effective and interoperable security management across physical and IT security systems. A focal point of the specifications is to improve the auditability of systems. The Open Security Exchange (www.opensecurityexchange.com), founded by Computer Associates, HID Corporation, Gemplus, and Tyco, was created to address today's lack of integration between various components of security infrastructures. See: www.opensecurityexchange.com

This table is (U//FOUO)

2.7.3.8.5 (U) Costs/Limitations

(U) The development of standards needed to provide a GIG-wide common log format, and aggregation and normalization scheme—as well as interoperable standards—might prove to be difficult and costly. Industry working groups such as the CIDF and IDMEF mentioned earlier have been stymied in the process of unifying and standardizing formats and information exchanges among disparate systems.

(U) The progress of these groups and initiatives is still tentative. The difficulty likely arises from political battles that affect current SEM vendors who have captured niche markets based on their point solutions and custom middleware. Standardization in the recording, storage, collection, analysis, monitoring and reporting phases will increase competition among these SEM vendors for each of these phases. Thus, there is little incentive for existing vendors to conform to component and application-based standards that would result in sacrificing niches in the SEM market to competition. However, as pointed out earlier, these limitations have to be overcome or at least reduced in order to provide a GIG-wide automated auditing solution.

2.7.3.8.6 (U) Dependencies

(U) A GIG-wide unified and automated audit technology solution will strongly depend on overcoming the limitations described earlier, and the advancement and adoption of standards-based recording, collecting, and monitoring solutions.

2.7.3.8.7 (U) Alternatives

(U) The alternative to utilizing automation with audit management is to use manual methods – which is not a viable solution. Manual methods and paper trails and have proven to be tedious, inefficient and unreliable. With the advent of smarter and faster-acting attacks, there is the need for immediately detecting deviations from normal operations. This includes especially the detection of zero-day attacks. Automating the four phases of audit management lifecycle appears to be the prudent approach.

(U) The alternative to adopting a unified standards-based GIG technological solution is to select individual SEM and middleware solutions for various needs. In fact, this is the modus operandi in today's commercial enterprises. The obvious disadvantage with this solution is the dependency reliance on the vendor to provide a holistic solution.

2.7.3.8.8 (U) Complementary Technologies

(U) Collection and Analysis-based technology standards at the back end, such as those found commonly in IDS (Intrusion Detection Systems) and IPS (Intrusion Prevention Systems) in particular, should complement the development of audit-analysis and audit-collection technology standards. Also, dynamic policy technology standards at the front-end should complement audit-recording technology development.

2.7.4 (U) Management of IA Mechanisms & Assets: Gap Analysis

(U//FOUO) Table 2.7-11 summarizes the adequacy of the technologies to meet the needs of this IA Enabler.

14193 **Table 2.7-11: (U) Technology Adequacy for Management of IA Mechanisms and Assets**

This table is (U//FOUO)

		Technology Categories								
		Identity Management	**Privilege Management**	**Key Management**	**Certificate Management**	**CM of IA Devices and Software**	**Inventory Management**	**Compromise Management**	**Audit Management**	**RCD Attributes**
IA Attributes	**GIG Identity Management**		N/A	N/A	N/A	N/A	N/A	N/A	N/A	IAIR1. IAIR2, IAIR3, IAIR4, IAIR5, IAIR6, IAKCM40, IAUAM1-IAUAM3
	GIG Authorization and Privilege Management	N/A		N/A	N/A	N/A	N/A	N/A	N/A	IAAM1, IAAM2, IAAM3, IAAM4, IAAM7, IAAM8, IAAM9, IAAM11, IAKCM41
	Policy based Access Control	N/A	N/A	N/A		N/A	N/A	N/A	N/A	IAAM12
	GIG Remote IA Asset Management	N/A	N/A	N/A		N/A	N/A	N/A	N/A	IANMA1
	OTN Benign Fill	N/A	N/A			N/A	N/A	N/A	N/A	IAKCM7, IAKCM9
	IA Asset Inventory Management	N/A	N/A	N/A	N/A			N/A	N/A	IAPS2, IAPS3, IANMA1, IANMA2
	Assured IA Asset Configuration Management	N/A	N/A		N/A		N/A	N/A	N/A	
	IA Asset Compromise Management	N/A	N/A	N/A	N/A	N/A	N/A		N/A	IAKCM35
	IA Asset High Robustness	N/A	N/A	N/A	N/A				N/A	

This table is (U//FOUO)									
	Technology Categories								
	Identity Management	**Privilege Management**	**Key Management**	**Certificate Management**	**CM of IA Devices and Software**	**Inventory Management**	**Compromise Management**	**Audit Management**	**RCD Attributes**
GIG Key Management	N/A	N/A		N/A	N/A	N/A	N/A	N/A	IAKCM1-IAKCM9, IAKCM12-IAKCM17, IAKCM24-IAKCM27, IAKCM33, IAKCM34, IAKCM36-IAKCM38
GIG Cert Management	N/A	N/A			N/A	N/A	N/A	N/A	IAKCM18, IAKCM19, IAKCM23, IAKCM24, IAKCM27, IAKCM39, IAKCM40, IAKCM43-IAKCM52, IANRP6
GIG Coalition Key Management t	N/A	N/A			N/A	N/A	N/A	N/A	IAKCM29, IAKCM30, IAKCM53
GIG Package Management									IAKCM32
GIG Management Auditing	N/A		N/A	N/A	N/A	N/A	N/A		IAIR5, IAAM11, IAKCM28, IAEM23, IANMP4
GIG Audit Logging and Analysis	N/A	N/A	N/A	N/A	N/A	N/A	N/A		IAIAC7, IAAUD1-IAAUD10
GIG CM Management	N/A	N/A	N/A	N/A			N/A	N/A	IACM1-IACM5, IASA05
This Table is (U//FOUO)									

2.7.4.1 (U) Identity Management

(U//FOUO) Provisioning and Maintenance Standards – Currently SPML is the only standard for provisioning, and there are no major standards for ongoing maintenance operations. SPML is a relatively new standard that needs wider adoption before it can stabilize. A maintenance standard needs to be developed to allow disparate aspects of an identity management enterprise to manage existing identities. These standards are required for an identity management deployment at the DoD level to support GIG activities. Once developed, these standards need to be integrated into new and existing identity management products and services.

(U//FOUO) Federated Identity – While there are some commercial early adopters of Federated Identity systems, this technology is still immature. Creating federated identity systems require a great deal of trust, coordination, and development between two federated partners. The current commercial model will likely not meet the DoD's requirements for security and dynamic administration. DoD-specific standards and guidelines need to be developed to support Federated Identity Management within the GIG.

(U//FOUO) GIG-Specific Identity Management Schema – When implementing an identity management system, a schema describing users, their properties, and profiles must be created. This schema can vary dramatically from enterprise to enterprise. For the GIG, a schema should be developed that encompasses DoD-wide needs. Further, systems need to be designed to handle potential future schema modifications. Whatever identity management schema is developed in the near term will likely need revision after a few years of deployed use.

2.7.4.2 (U) Privilege Management

(U///FOUO) There is a standard that defines an Attribute Certificates to bind privileges to an Identity Certificate. This standard is an extension of the X.509 standard and has been adopted widely by PKI. Today, PMI works within the PKI infrastructure. Scalable alternatives to Attribute Certificates need to be explored.

(U//FOUO) There are limitations in the capabilities currently provided by PMI.

(U///FOUO) There are no standard mechanisms that specify how privileges are to be managed in a RAdAC Model that is required by the GIG.

(U///FOUO) Another gap is the lack of technologies and standards that accommodate MLS classifications. This is likely a policy gap as well.

(U///FOUO) Furthermore, while privileges for individuals are accounted for within existing PMI, standards and formats that address dynamically changing communities (COI) or Role-based privileges need to be developed and standardized across the GIG enterprise.

(U///FOUO) Finally, policies on the trusted transportation and distribution need to be developed GIG-wide as well.

2.7.4.3 (U) Key Management

(U//FOUO) Automated solutions for managing the life cycle of keys do not currently exist. Human intervention is required in many aspects of key management, including registration, distribution, revocation, re-keying, and destruction. These human access points are vulnerable to threats and errors. To mitigate these vulnerabilities, there needs to be a strong drive towards standards for automation that provide and control the management of the life cycle keys. One such identified initiative that is driving requirements and standards is the KMI effort. The outcome of the KMI effort is expected to produce standards and policy.

(U//FOUO) The identified gap areas within the individual aspects of key management include both policy gaps and technological gaps.

(U//FOUO) One major gap area is the weakness or non-existence of tying policy controls (including dynamic policy changes) to various aspects of the key management cycle in an automated fashion. Standards need to exist so that automation can be built into promulgating dynamic policy changes into the necessary rules and regulations with which key registration, packaging, distribution, re-keying, revocation, and destruction work seamlessly and in an up-to-date, situational, manner.

(U//FOUO) Another gap area is the lack of standards for unified key labeling, packaging, and distribution formats. The only area where some semblance of standards exist here are in the PKI (public, asymmetric keys) infrastructure. But none exists beyond PKI. Moreover, PKI has its own limitations with keys—such as re-keying—since PKI is certificate driven and not so much key-driven. In the Type-1 Classified arena, the key packaging and distribution processes are mainly manual processes. While they follow individual and situational-based policy, there are no standards to unify these in order to eliminate or reduce manual error-prone and human access vulnerabilities towards threats. Standards and technologies should include the incorporation of MLS systems and data stores to close these gaps.

(U//FOUO) The management of symmetric keys needs to be included and evolved as well. For example, while there are individually controlled escrows and distribution of symmetric keys, there are no identified standards for the unified distribution of keys that would be required in the GIG-wide enterprise.

2.7.4.4 (U) Certificate Management

(U//FOUO) The only existing Certificate Management standard that exists today is found in the PKI arena. However, PKI has interoperability limitations at the application and component levels. There are no identified interoperability standards or technologies that specify the interfaces for certificate and data exchange between CAs. There are secure transports currently in use for certificates, but as such, there is no GIG-wide enterprise policy that governs what these access control restrictions should be.

(U//FOUO) There are standards that are supposedly emerging for enhancing certificate attributes that aim to capture additional significant information such as subject privileges, trust anchor information and other necessary identity, trust, distribution, and access control information. There are also initiatives that are attempting to specify and collate various levels and classifications of certificates such as the Class 3, Class 4, and Class 5 Government and commercial type certificates. Until that happens, there is a gap here.

(U//FOUO) There are standards and technology gaps in the manner that the GIG would require cryptographically binding, public keying material to information and attributes associated with a particular user/entity using a trust anchor in order to certify that the private key corresponding to the public key in the certificate is held by the same user/entity. There is a need to have binding strength increase with the strength of the cryptographic algorithm and key length used. No such standards or policy exist today.

2.7.4.5 (U) Configuration Management of IA Devices and Software

(U) Commercial products do not currently address a number of GIG requirements:

- (U//FOUO) Although some configuration management tools authenticate the server, few use encrypted communications channels
- (U//FOUO) Authentication of client machines is nonexistent
- (U//FOUO) Only one identified product provides any support for modeling configuration changes before deployment
- (U//FOUO) Although many products provided support for test deployments before a patch or upgrade deployment, none provide support for testing the configuration
- (U//FOUO) No product provides support for authenticated or cryptographic verification of configurations, all assumed the device configuration information could be trusted
- (U//FOUO) No product provides support for sensitive material distribution, such as keys, which require protection, receipts, and auditable tracking of delivery
- (U//FOUO) No product supports remote update of firmware.
- (U//FOUO) No general standard exists for communications between a configuration manager and its agent, or the target machines, although Microsoft has implemented the WBEM standard for its operating systems.

2.7.4.6 (U) Inventory Management

(U) The technology gap for Inventory Management lies in the area of RFID technology. Although the technology has been available since 1974, standardized, interoperable tags and readers are a recent innovation. Although the field is rapidly developing on its own, the focus is on developing low-cost, passive UHF RFID tags that reach the critical $.05 per unit goal that can be used for consumer supply chain applications. Consumer privacy issues have raised the concern that RFID tags are still active after leaving the retail sales point and could be used to track individuals, so some work is being done to develop RFID tags that can be killed—rendered permanently inert or unresponsive to a reader interrogation, or rendered temporarily inert. However, no apparent work is being done to develop secure RFID tags which would respond only to interrogation and commands from an authenticated reader. Even DES encryption is considered significantly more expensive than can be handled by an RFID tag.

(U) The greater gap for Inventory Management is that it requires development of an Inventory Management infrastructure that tracks and manages.

2.7.4.7 (U) Compromise Management of IA Devices

(U) Tamper detection mechanisms are well understood, although tamper resistant mechanisms such as seals can always be defeated. However, current systems limit their tamper response to zeroizing their internal data and do not include the concept of network-aware reporting of alerts—secure or otherwise—as part of their tamper processing.

(U) External compromise monitoring mechanisms such as an IAC status and monitoring protocol or IAC Keep Alive protocol do not currently exist, and must be developed. It could become part of a SNMP Management Information Base or part of an IA device management protocol. A secure device management protocol is a requirement brought by secure configuration management requirements.

2.7.4.8 (U) Audit Management

(U//FOUO) Audit management exists today in a very non-standard manner. There are a multitude of SEM vendors that provide some type of audit capability built into their proprietary solutions. Without standards in technology, interoperability (within components, log formats, audit analyses, etc,), and policies, there is a big gaping hole in the unified audit management scheme that the GIG enterprise requires.

(U) Standards in technology, interfaces, interoperability, and policies need to be developed and defined in the areas of:

- (U//FOUO) Log and event formats – to capture and record normalized GIG-wide activities and system performance
- (U//FOUO) Standardized Securing of audit data into one-way (diode) stores
- (U//FOUO) Standardizing Agents and Agentless components for interoperability and security (assurance)
- (U//FOUO) Standards for tools that monitor system resources

- (U//FOUO) Adhering to the DoDI 8500.2 standards for audit data record capture. This includes provisioning for attributes, such as the ECAR-1, 2 and 3, that correspond to the various classification levels
- (U//FOUO) Central monitoring and interfacing standards, from a NOC or SOC
- (U//FOUO) Standards for correlation, analysis and alerting services that subscribe to audit data publishing
- (U//FOUO) Secure transport standards.

2.7.5 (U) Management of IA Mechanisms and Assets: Recommendations and Timelines

(U) The management of network assets itself is relatively mature, however, this is true only for low-threat environments. In a medium to high-threat environment, a significant gap exists. For the high-assurance management of cryptographic components, there are only limited proprietary solutions. No solutions exist which provide configuration management of high assurance IA devices.

2.7.5.1 (U) Standards

(U) Standards that need to be developed to support the management of GIG Assets and Mechanisms include:

- (U//FOUO) Standard for maintenance, communication, and management of existing identities across federated authorities
- (U//FOUO) DoD-specific standards and protection profiles for federated identity management, including a DoD-wide identity management schema
- (U) Secure device identification standards, which use cryptographic authentication of the identity of a device.
- (U) Standards for dynamic establishing and disestablishing COIs and COI membership
- (U) Standards for role-based privilege management across federated organizations
- (U) Standards for wholly automated life cycle for key material
- (U) Standards for key labeling, packaging, and distribution, particularly symmetric keys
- (U) Standards for interoperability among certificate management infrastructure components
- (U//FOUO) Standard protocols for the secure management of IA-enabled devices, including initialization, software load, configuration, verification of a configuration, and update
- (U//FOUO) A standard for secure boot and remote initialization of a device, including device authentication; especially a cryptographic device across a black network

- (U) Standards for secure remote data delivery including receipting
- (U) Secure RFID standards
- (U//FOUO) Standard IAC keep-alive protocol
- (U//FOUO) Standard compromise notification protocol, particularly in the case of notification across a black network
- (U) Widely adopted audit log standard
- (U) Audit aggregation and analysis data standard.

2.7.5.2 (U) Technology

- (U) Secure, authenticated network boot devices
- (U) Secure RFID, including authentication of the reader to the RFID tag
- (U) Tamper detection and network manager notification
- (U//FOUO) Multi-level PKI certificate authorities for a single identity and certificate across the GIG.

2.7.5.3 (U) Infrastructure

- (U) Device identification and tracking
- (U) IA device inventory and configuration management
- (U) Key Management Infrastructure.

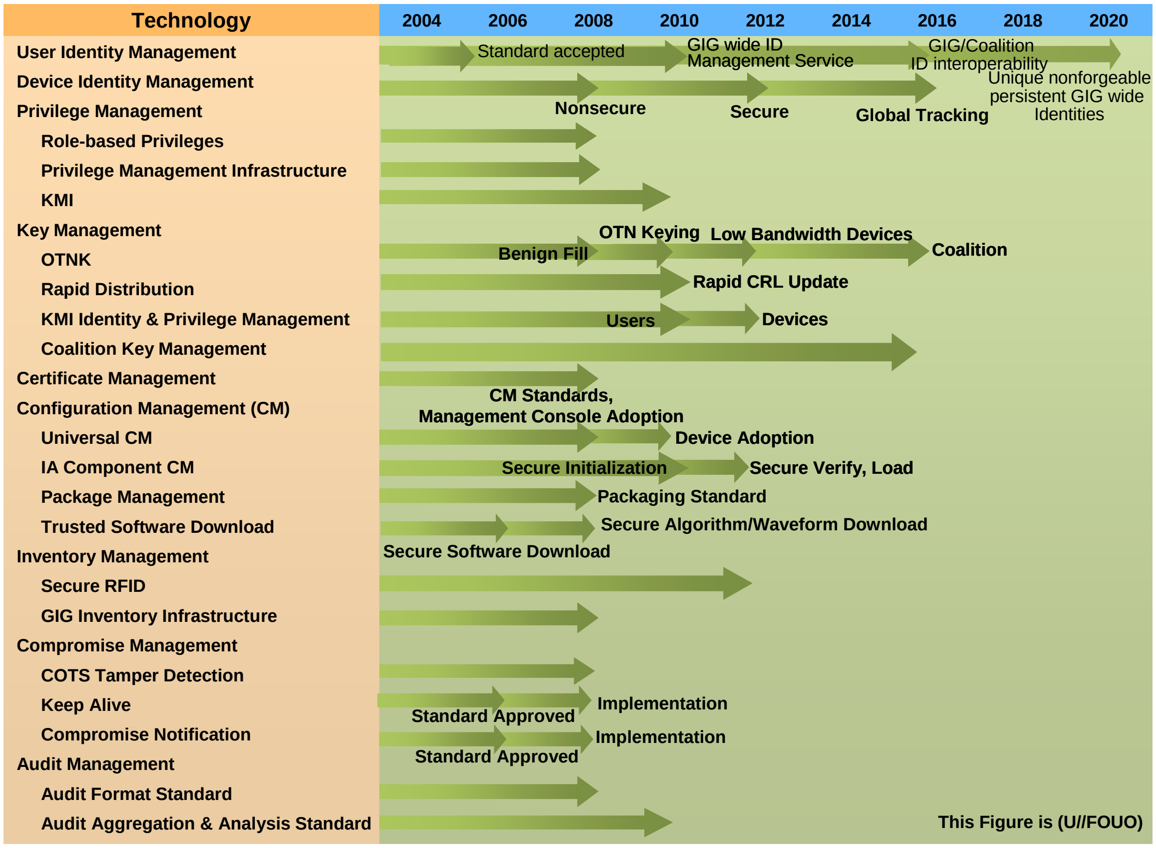


Figure 2.7-16: (U) Technology Timeline for Assured Resource Allocation

3 (U) SUMMARY

(U//FOUO) The Global Information Grid (GIG) Information Assurance (IA) Capability/Technology Roadmap compares the commercial and Government technology trends and technology forecasts available today against the needed capabilities defined in the Transition Strategy in the GIG IA Reference Capability Document (RCD). The results of these analyses include descriptions of interdependencies between needed capabilities, technology timelines, and gaps between capability needs and technology availability. These results, together with other background information and analysis in this document, are intended to provide decision makers with the information needed to revise or write new standards and policies, develop implementation guidelines, make research funding decisions, devise strategies for needed technology development, and develop technology implementation plans.

(U//FOUO) This section summarizes the most significant impressions and conclusions arising from the investigations and analyses of the candidate IA technologies. Results are organized around the four IA cornerstones defined in the GIG IA RCD and presented in the context of the Transition Strategy. The four IA cornerstones are:

- (U) Assured Information Sharing
- (U) Highly Available Enterprise
- (U) Assured Enterprise Management and Control
- (U) Cyber Situational Awareness and Network Defense

(U) Some of the technologies support more than one cornerstone. Therefore, results for any particular technology may appear to be duplicated in two or more cornerstones. However, there are generally slight differences in the gaps and recommendations, reflecting the different aspects of the cornerstone that the technology supports.

(U//FOUO) For each IA cornerstone, a summarizing timeline is shown that illustrates the primary technology categories described in the Transition Strategy and needed to meet 2008 GIG IA capabilities. Gaps and recommendations are then described for the technology areas and component technologies, where appropriate. In the timelines, milestones for specific imperatives are shown as colored diamonds, where:

- (U) Green indicates that the milestone will be achieved under current development plans, schedules, and funding of the component technologies supporting that milestone
- (U) Yellow indicates that the milestone will not be achieved if development of the supporting technologies proceeds as planned—but the milestone could be achieved by accelerating current development efforts or starting new development efforts
- (U) Red indicates that the milestone cannot be achieved as currently defined by the Transition Strategy

(U) The milestone color-coding is largely based on isolated examinations of the supporting component technologies. In practice, some technology development efforts will be interdependent. For example, one technology development effort may be delayed because it must rely on an intermediate result from another technology development. Such interdependencies were not fully considered, so some of the technology development timelines and the color-coding of the affected milestones may be slightly optimistic. Further investigation will be needed to refine these timeline estimates.

(U) With only minor exceptions, the gaps and recommendations are described for technologies needed to meet the 2008 GIG IA objectives as described in the Transition Strategy. The description is further limited to technologies that are deemed risky, either because no work is currently going on, or because ongoing development effort will probably not be completed in time to deploy for 2008. In some cases, gaps and recommendations are summarized for technologies needed for 2012 and beyond, but only in cases where technology development efforts must begin now in order to meet those technology milestone dates.

(U) These results give a fairly complete picture. Subsequent effort on this document will focus on updating and refining the status of the technologies.

3.1 (U//FOUO) ASSURED INFORMATION SHARING SUMMARY

(U//FOUO) Technologies supporting this cornerstone are organized into five general categories: Identification, Authentication, Access Control, Data Labeling, and Cross-Domain Security.

(U) Figure 3.1-1 provides an overview of the technologies and how they support the IA imperatives listed in the Transition Strategy. As shown, while none of the technologies will be completed in time to meet the 2008 IA imperatives, the milestones are achievable if current efforts are accelerated. Some imperatives have no supporting technologies identified in this release of the document. These are discussed in the gaps below.

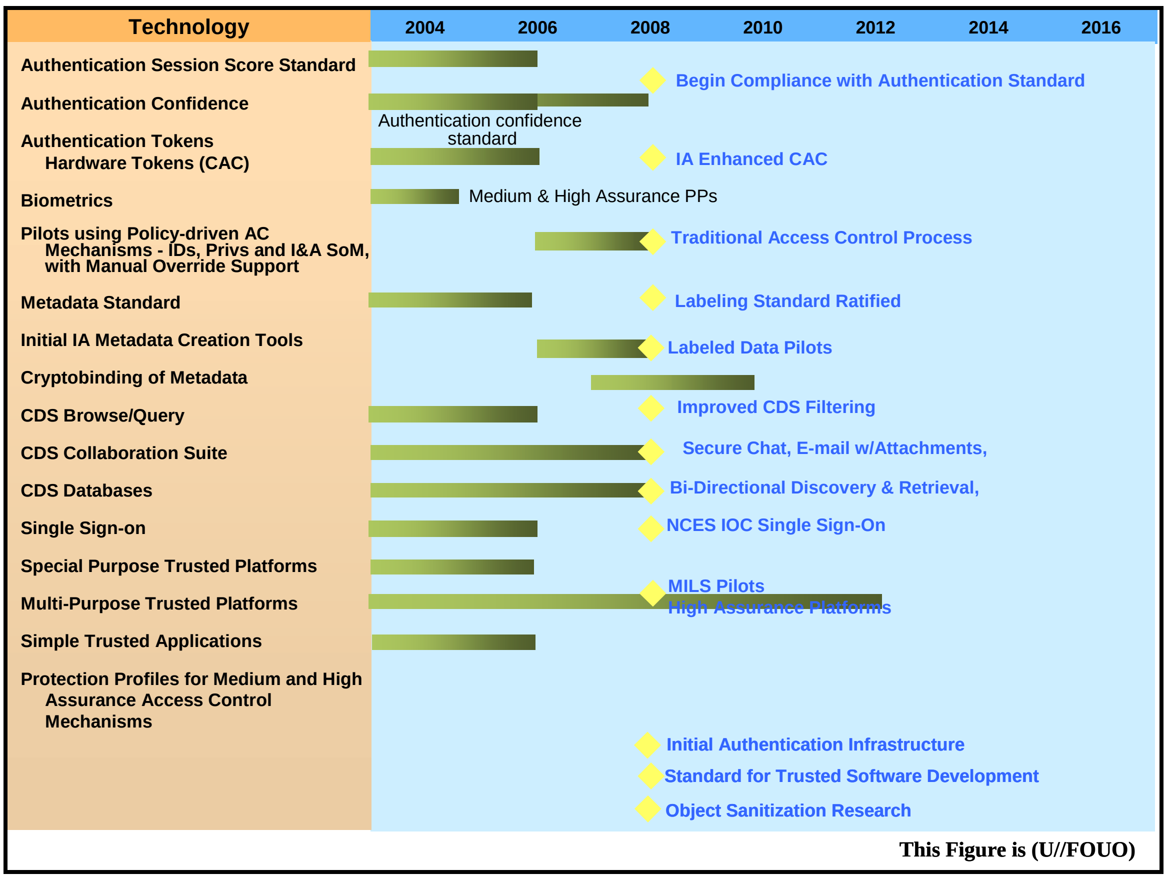


Figure 3.1-1: (U//FOUO) Technology Timeline for Assured Information Sharing

3.1.1 (U) Identification and Authentication Technologies

(U//FOUO) As the technology development efforts currently stand, none of the I&A-related milestones shown in Figure 3.1-1 will be met. Gaps that will prevent deploying an initial authentication infrastructure that conforms to a common authentication standard are listed below—along with recommended corrective actions.

- (U//FOUO) Gap: An authentication framework standard does not yet exist. Such a standard (or set of standards) must address SoM levels, authentication session scoring, a SoM forwarding structure, and authentication confidence metrics. Until such standards exist, a global authentication infrastructure and associated technologies cannot be deployed.

  (U//FOUO) Recommendations: 1) Develop a common GIG-wide device/service authentication techniques and standards.

  2) (U//FOUO) Rapidly advance research into the relatively new area of authentication confidence metrics.

- (U//FOUO) Gap: Protection Profiles are needed for Medium and High Assurance authentication technologies, including biometrics technologies.

  (U//FOUO) Recommendations: Develop protection profiles to facilitate authentication standards and architecture development.

- (U//FOUO) Gap: A common GIG-wide Single Sign-On (SSO) mechanism, protocol, and architecture have not yet been selected.

  (U//FOUO) Recommendation: Study and select a GIG-wide architecture for SSO using the candidate approaches described in Section 2.1. Include in this study a complete analysis of the proposed NCES SSO architecture.

- (U//FOUO) Gap: A scalable authentication server that is able to interpret and use I&A session scores and comply with the GIG authentication standards does not exist.

  (U//FOUO) Recommendation: Begin development of a scalable, robust, and distributed authentication server capability whose components can operate in multiple architectural constructs (e.g., in-line, embedded, coprocessor, remote).

(U//FOUO) In addition to the technologies listed above, other gaps have been identified that will prevent meeting 2012 (and later) imperatives. Those listed below require that recommendations be acted on soon in order to ensure sufficient development time to meet the affected milestones.

- (U//FOUO) Gap: A high assurance DoD PKI Class 5 token with Type I cryptography will eventually be needed. Development of the DoD CAC is proceeding in the needed direction, but it is not yet available. A Class 5 token will be needed for assured access to classified information. Such a token will use Type I cryptography, and its security-critical functionality will be assured throughout its life cycle, including design, development, production, fielding, and maintenance.

  (U//FOUO) Recommendation: Monitor ongoing and future developments of the DoD CAC to ensure support of all future GIG requirements (including the Class 5 token).

- (U//FOUO) Gap: Common standards for Partner Identity Proofing and a common Identification Registration/Management Infrastructure will be needed to ensure identity interoperability among all current and future GIG partners (e.g., DoD, IC, civil Government, Department of Homeland Security (DHS), allies, coalition partners).

(U//FOUO) Recommendation: Begin formation of future partner community; then start development of a common Partner Identity Proofing standard.

3.1.2 (U) Access Control and Data Labeling Technologies

(U//FOUO) The Risk Adaptable Access Control (RAdAC) functions central to GIG access control are in their infancy with respect to concept formulation, standards development, policy implications, and technology implementation. While industry has shown interest in role-based access control, and now attribute-based access control, the unique features of RAdAC require additional technologies.

(U//FOUO) Moreover, industry is not likely to sponsor the needed research and development in this area, since no commercial market is anticipated for such a capability. Therefore, there are numerous technology gaps that the Government will need to address. Only the first gap listed below is called out in the 2008 Transition Strategy imperatives. The remainder can and should be closed by 2008 in order to meet the imperatives of subsequent increments.

- (U//FOUO) Gap: Protection Profiles. There are no current or planned protection profiles that address RAdAC or attribute-based access control. These protection profiles are necessary to establish the minimum security protections required for any implementation of RAdAC.

  (U//FOUO) Recommendation: Develop Attribute-Based Access Control (ABAC) and RAdAC Protection Profiles.

- (U//FOUO) Gap: RAdAC standard. Since industry is not moving in the RAdAC direction, there are no formal representations of architecture, interface definitions, performance requirements, or protocol requirements.

  (U//FOUO) Recommendations: Develop a RAdAC standard. Also, begin RAdAC prototyping to support standards development. This activity will also be valuable for other related RAdAC development activities, including requirements discovery, input ontology development, Digital Access Control Policy (DACP) standard development, and Digital Rights integration specification development.

- (U//FOUO) Gap: ABAC standard. Given the current immaturity and criticality of RAdAC, it would be prudent to have an alternative to RAdAC. ABAC should be considered as an interim solution while RAdAC is being developed. However, even though there is research and even commercial ABAC-based products, there are no commercial or government standards.

  (U) Recommendation: The Government should initiate development of a commercial or government ABAC standard.

- (U//FOUO) Gap: RAdAC mathematical model: An underlying mathematical model is needed to meet Medium and High assurance implementation requirements and to assist in the transformation from a Discretionary Access Control (DAC) and Mandatory Access Control (MAC) access control culture. This model needs to include the digital access control policy since the two are so tightly integrated.

(U//FOUO) Recommendation: Develop RAdAC mathematical model.

- (U//FOUO) Gap: Input parameter ontology. All attributes that feed the RAdAC model need to have an ontology that is accessible and standardized. This applies to attributes of IT Components, Environment, Situation, Soft Objects (metadata), and people.

    (U//FOUO) Recommendation: Develop input parameter ontology.

(U//FOUO) There is at least one ontology standard and language that meet some of the basic requirements for DACP. However, significant work is needed to realize a complete implementation that will meet GIG information-sharing requirements.

- (U) Gap: DACP standard. Based on the underlying math model, a DACP standard that uses ontology and deontic languages needs to be developed. This standard will address the access control policy grammar, exception handling, business rules about allowable and disallowable policy constructs, and business rules for policy negotiation and de-confliction.

    (U) Recommendation: Develop DACP standard with associated business rules.

- (U) Gap: Digital Rights Management integration specification. Digital Rights can be viewed as a static projection of digital access control policy onto a particular soft object. There is currently ongoing research in the Digital Rights realm and proposed standards, but none of this work is aimed at specifying a relationship between digital rights and digital access control policy. An analysis of these relationships, digital rights implementation, and Policy Enforcement Point interface is necessary to complete the end-to-end access control of GIG information and support the transition to a need-to-share culture.

    (U) Recommendations: 1) Develop Digital Rights integration specification

    (U) 2) Work with commercial standards groups to integrate needed aspects into the appropriate commercial standards.

(U//FOUO) The RAdAC core technologies present the most technical risk for access control, but gaps in metadata technologies are also of concern because of the centrality of metadata to assured information sharing. These gaps can and should be closed by 2008 in order to meet the imperatives of subsequent increments.

(U//FOUO) Each data object will be associated with a Quality of Protection (QoP) that specifies how that object is to be protected while at rest, and how it is to be protected throughout its lifetime. This impacts the technology employed and design of virtually every entity in the GIG that handles data.

- (U//FOUO) Gap: The definition, implementation, and enforcement of QoP at the data object level.

    (U//FOUO) Recommendations: A QoP standard must first be developed that defines the privileges that can be assigned to each data object. Analytical and modeling-based studies will be needed to develop appropriate policies, standards, and specifications for all affected entities.

- (U//FOUO) Gap: Standards. Both the Intelligence Community (IC) and Department of Defense (DoD) are developing metadata standards, and they are coordinating their work to ensure that IA attributes associated with RAdAC style access control decision-making and discovery are addressed in these standards. However, standards development activities must be closely coordinated with ongoing research and development efforts, in order to avoid incompatibilities in technology standards that would eventually require changes to supporting tools, infrastructure, and large quantities of existing metadata records.

  (U//FOUO) Recommendations: 1) The GIG community should work with IC and Core Enterprise Service (CES) Metadata working groups to ensure IA RAdAC required attributes are adequately addressed, and to either guide the integration of IA attributes into the metadata standards according to detailed analysis, or (preferred) support the merger of these standards.

  2) (U//FOUO) Before stabilizing the metadata standards and IA attributes, conduct further studies to examine the impact of metadata on network traffic/overhead (especially for real-time and session object types) and potential for trading metadata IA granularity with transmission overhead.

- (U//FOUO) Gap: Metadata creation tools. Commercial metadata creation tools are available. However, they do not have the needed GIG IA-related capabilities and interfaces, which are new, complex, and unique to the GIG.

  (U//FOUO) Recommendation: Begin now early design of metadata creation tools in parallel with the metadata standards definition to ensure IA specific attributes, cryptographic binding of metadata and the source object, and authorization interface needs are addressed.

3.1.3 (U) Cross-Domain Technologies

(U//FOUO) Despite the large number and variety of Cross Domain Solution (CDS) development efforts underway for many years and moderate number of accredited products available, significant work remains in order to meet the CDS-related 2008 GIG IA requirements of the Transition Strategy.

- (U//FOUO) Gap: Cross-domain file transfer. Although accredited solutions exist to transfer fixed-file formats, there are many files prohibited from being passed through these solutions. Most notably these include executable files and documents with macros—Microsoft Office files in particular.

  (U//FOUO) Recommendations: 1) Research and develop advanced capabilities for safely transferring files across security domains, initially targeting the examination of files generated by Microsoft Office and other common warfighter applications for executable and hidden malicious content.

  2) (U//FOUO) Develop clear and consistent policies for dealing with discovered malicious content, such as automatic deletion of content, imposition of execution constraints, manual security review, etc.

3) (U//FOUO) Develop mechanisms to execute the malicious content discovery policy. 4) Investigate alternatives to commonly-used products known to contain security weaknesses in this area.

- (U//FOUO) Gap: Trusted workstations, needed to push multiple domain access out to users in the field and support warfighter applications in the operational environments, are not available.

  (U//FOUO) Recommendations: Accelerate research to develop trusted CDS platforms that are:

  1) (U//FOUO) certified to allow users who are not cleared for the highest levels of information on the workstation to use the platform at the level for which they are cleared;

  2) (U//FOUO) allow warfighters to use applications to which they are accustomed, e.g., for word processing, collaboration, situational awareness, and planning;

  3) (U//FOUO) can function under the resource constraints of the warfighters (e.g., space, weight, and power constraints of infantry) while supporting critical functionalities (e.g., combat ID, secure voice).

- (U//FOUO) Gap: Information protection technologies (e.g., High Assurance Internet Protocol Encryptor [HAIPE]) supporting the GIG Black Core concept are currently single security domain devices and prevent traditional CDS from examining information flow content.

  (U//FOUO) Recommendation: Enhance functionality of data protection technologies to support information flows between security domains. Tighter integration between the content review and filtration system (e.g., the high assurance guard), and the protection system (e.g., the HAIPE) is required.

- (U//FOUO) Gap: Current cross-domain solutions are designed to examine static blocks of information containing entire message sets (e.g., files, email), and no ability currently exists to support critical real-time information flows (e.g., secure voice, video teleconferencing).

  (U//FOUO) Recommendation: Efforts for developing technologies to support cross-domain real-time flows—such as voice communications and collaboration among coalition partners—should begin immediately.

- (U//FOUO) Gap: Current maturity of IA controls has resulted in cross-domain solutions with strict management and configuration properties that do not facilitate flexible management, configuration, and adaptation of the CDS to insure proper operation in a changing environment (e.g., INFOCON transitions, dynamic multinational agreements, etc.).

  (U//FOUO) Recommendation: Develop standards, techniques, and procedures that can be certified to insure that CDS initialization, management, configuration and support shall not be impaired by use in remote warfighting environments among Joint and Multinational participants with dynamic agreements.

- (U//FOUO) Gap: Insufficient training and inadequate deployment of a Joint cross-domain solution leads to ineffective use of existing Service-owned CDS capabilities, restricts the flow of vital information, and complicates the correlation of information from multiple security domains.

  (U//FOUO) Recommendation: Develop standards for cross-domain technologies that are based on current Joint and Multinational operational doctrine and practices. These standards apply to the entire lifecycle of CDS technologies and include the development of common, Joint CDS capabilities, adequate deployment of Joint solutions; and sufficient training for the warfighters who will use these solutions.

3.1.4 (U) Trusted Platform Technologies

(U//FOUO) Trusted platforms have been around for more than 20 years in one form or another. For special purpose IA components, such as firewalls and gateways, the technologies are mature and will meet the 2008 GIG IA imperatives. For workstations and other devices that must connect to multiple security domains, significant research and development in the areas of software engineering, high-assurance computing, network security, and system evaluation will be required before needed GIG IA capabilities can be met. The primary gap and the action needed to meet 2008 GIG IA imperatives are:

- (U//FOUO) Gap: Software development for trusted applications. No universally-accepted methodologies—much less standards—have been devised for development of software to be used in applications requiring high assurance. This problem has been recognized, and Office Secretary of Defense (OSD) Networks and Information Integration (NII) and DHS are co-sponsoring an effort to investigate the problem of high-assurance software, with the goal of establishing partnerships between Government, academia, and industry to develop solutions that span the software development process, evaluation, and training. However, it is not clear if the current efforts will result in the publishing of standards for trusted software development by 2008.

  (U//FOUO) Recommendation: Given the importance of high-assurance software to GIG components, DoD should accelerate its current study efforts and focus on devising trusted-software development processes and standards.

- (U//FOUO) Gap: A linkage between a security policy enforced by the trusted application and the security policy enforced by the host platform needs to be developed. This is the composition problem that has been researched off and on with unsatisfactory results for at least 20 years. A side issue to be examined is what happens when the trusted application is implemented on a variety of host platforms, and those platforms must communicate and interoperate.

  (U//FOUO) Recommendation: Conduct research aimed at coordinating application security policy and hardware security policy.

- (U//FOUO) Gap: Construction of self-protecting applications that can guard themselves against attacks coming through the host platform, such as against attacks using disk storage or input devices.

(U//FOUO) Recommendation: Conduct research into trusted applications that can guard themselves against attacks coming through the host platform (hardware or software).

- (U//FOUO) Gap: Support for complex security policies within trusted platforms, such as dynamic access control policies like RAdAC.

  (U//FOUO) Recommendation: Conduct research aimed at defining and enforcing complex security policies with trusted platforms. Include research into developing and enforcing RAdAC policies.

3.2 (U) HIGHLY AVAILABLE ENTERPRISE SUMMARY

(U//FOUO) Technologies supporting this cornerstone are organized into five general categories: IA Policy-based Routing, End-to-end Resource Allocation, Edge-to-Edge Boundary Protection in the Black Core, Secure Voice, and Quality of Protection.

(U) Figure 3.2-1 provides an overview of the technologies and how they support the IA imperatives listed in the Transition Strategy. As shown, while none of the technologies will be completed in time to meet the 2008 IA imperatives, if current efforts are accelerated, the milestones are achievable.

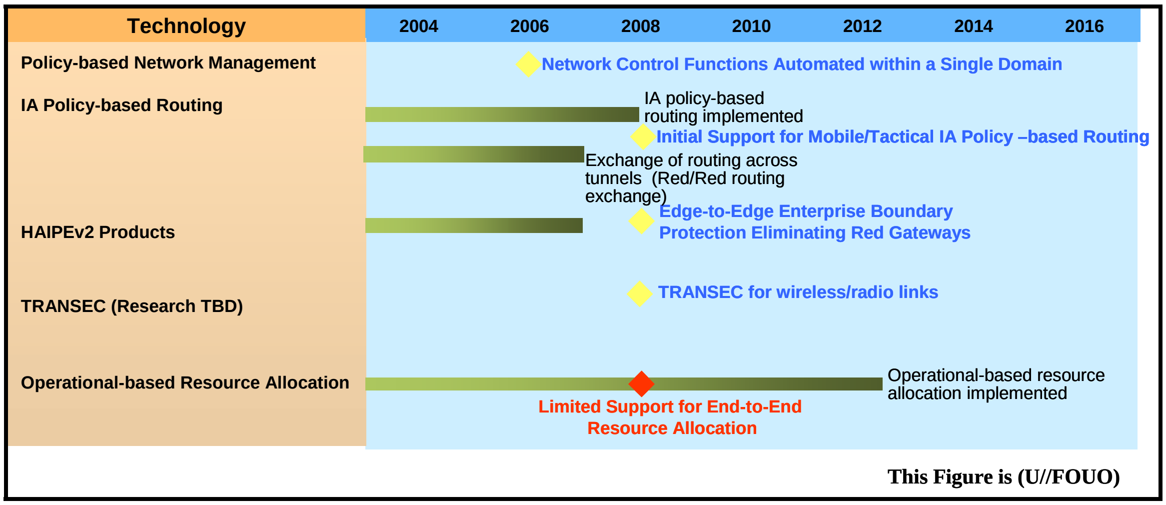


Figure 3.2-1: (U//FOUO) Technology Timeline for Highly Available Enterprise

3.2.1 (U//FOUO) IA Policy-based Routing for Mobile/Tactical Environments Technologies

(U//FOUO) As described in Section 2.5, all routing is policy-based, but a policy usually enforces the shortest path or least cost rather than considering the IA properties of the links. The extension of routing protocol algorithms to include the aspect or metric of path assurance/security is relatively recent and thus not nearly as mature. Some work in this area has been done in the area of IA policy-based routing for mobile ad hoc networks, due to the obvious potential vulnerabilities of wireless networks as compared with more secure wired network infrastructures. Research for IA policy-based routing needs to be continued.

- (U//FOUO) Gap: Lack of IA metrics in wired and wireless routing protocols.

  (U//FOUO) Recommendations: Further research and development of adaptive security-driven (i.e., IA policy-based), wireless routing algorithms is required for inclusion in mobile and tactical programs (e.g., Joint Tactical Radio System [JTRS] and Warfighter Information Network-Tactical [WIN-T]). Research should be extended into the wired domain so that IA policy-based routing can benefit all networks (wired or wireless). Findings must be used to advance the standards evolution and demonstration/implementation of extensible routing protocols (such as

Open Shortest Path First [OSPF] and Intermediate System-to-Intermediate System [IS-IS]) so that IA metrics can be fully employed in routing decisions.

3.2.2 (U) End-to-End Resource Allocation Technologies

(U//FOUO) Resource allocation traditionally has been limited to the scope of small geographic areas as opposed to the world-wide reach of the GIG. The GIG must be able to control and modify the amount of resources (e.g., bandwidth, processor cycles) allocated to any given user, based on current operational requirements. For example, the GIG should be able to cut back on the amount of resources available to sustain operations on portions of the network in order to increase the resources available to a unit currently engaged in battle.

- (U//FOUO) Gap: At this point, there is insufficient research—much less technology—to support all of the GIG requirements for dynamic resource allocation. Dynamic reconfiguration of resources is a difficult problem that has only some limited solutions available now.

  (U//FOUO) Recommendation: Initiate research into permitting dynamic reconfiguration within a Black Core where all traffic is encrypted while at the same time defending against attacks as needed in the GIG (e.g., ensuring that a requested change in resources comes from an authorized entity and is legitimate and appropriate given the current operational situation).

(U//FOUO) Part of resource allocation involves the deployment of Quality of Service (QoS) mechanisms across the GIG. While there has been a significant amount of work done by commercial industries related to QoS, implementing and enforcing QoS mechanisms has proven difficult. Commercial products are evolving to support QoS, and the GIG must keep abreast of new developments and integrate them where appropriate.

- (U//FOUO) Gap: An area of QoS that has not being given much attention by commercial industry is security mechanisms. QoS parameters need to be applied to packets and flows across the GIG by devices that do not abuse the features of QoS to use more than their share of resources or create Denial of Service conditions.

  (U//FOUO) Recommendation: Define the IA aspects of QoS and socialize them across the GIG community. Define procedures and mechanisms for end-to-end QoS and resource allocation across crypto boundaries. Define security mechanisms and solution for supporting end-to-end QoS in the GIG. Solutions need to be developed to support the end-to-end QoS GIG requirements.

(U//FOUO) QoS solutions are currently being deployed within the GIG. Although the Transition Strategy does not specify end-to-end QoS enforcement until 2012, research and development must continue in order to mitigate the risk of non-interoperable QoS islands within the GIG.

- (U//FOUO) Gap: An additional capability related to resource allocation that is not being considered by commercial industry is precedence and preemption in the Black Core. The GIG has requirements (particularly with regards to voice) to assign priority (different from QoS) to packets, and in times of congestion, higher priority packets can preempt lower priority packets.

(U//FOUO) Recommendation: Development of a GIG Precedence and Preemption standard to provide the capability for rational post-preemption rescheduling should continue so as to not leave GIG customers without requested services.

3.2.3 (U//FOUO) Edge-to-Edge Boundary Protection Technologies

(U//FOUO) GIG programs need to provide boundary protections without the use of red gateways. Within the Black Core, traffic will be encrypted at the boundary of the originating network and remain encrypted across the GIG transport programs until it is decrypted at the ingress to the recipient's network.

(U//FOUO) Gap: Traditional firewalling, content filtering, intrusion detection, and other IA capabilities will not function in the Black Core as they need to do today. The GIG community still has a need for these IA capabilities in the Black Core.

(U//FOUO) Recommendations: Resolving these issues will require research and testing as well as significant community socialization to ensure that solutions are consistently applied across the GIG and end-to-end services can be supported. Specifically the following areas need to be addressed:

1. (U//FOUO) Evolution of the HAIPE protocol is required to support dynamic routing in a multi-homed environment, red-to-red routing exchanges, QoS, dynamic black IP addresses, mobility, end-system implementations, resource-constrained implementations, and low-bandwidth, high bit error rate environments
2. (U//FOUO) Research is necessary to enable filtering on source, destination, and payload in the Black Core in order to monitor for unauthorized traffic before it crosses a GIG network
3. (U//FOUO) Research is necessary to provide admission control and priority handling of encrypted packets
4. (U//FOUO) Research is necessary to develop effective intrusion detection capabilities on encrypted segments.

3.2.4 (U) Secure Voice Technologies

(U//FOUO) Based on the 2008 Transition Strategy, Voice over IP (VoIP) solutions will be deployed within system high networks. While this is achievable with today's technology using a single vendor's solution, much work is required to move towards interoperable secure voice over secure IP solutions required by the GIG 2020 Vision.

- (U//FOUO) Gap: Lack of interoperable secure voice over secure IP solutions.

(U//FOUO) Recommendations: Activities that must be started to achieve the GIG 2020 Vision related to voice include:

1) (U) Standards for providing interoperability between Secure Voice over IP systems and Voice over Secure IP systems

2) (U//FOUO) Standards defining a common interoperable implementation of Future Narrow Band Digital Terminal (FNBDT) over IP networks, including call control, gateway operation, and user media details

3) (U//FOUO) Standards defining FNBDT multipoint operation (conferencing, net broadcast, and multicast applications)

4) (U//FOUO) Standards defining additional voice coders for FNBDT systems on specific GIG sub-networks

5) (U//FOUO) Interoperability between secure voice products in circuit switched networks and secure voice products in packet switched networks.

3.2.5 (U) Enforcement of QoP in Transit Technologies

(U//FOUO) Each data object will be associated with a QoP that specifies how that object is to be protected and routed across the GIG. This impacts the technology employed and design of virtually every entity in the GIG that handles data.

- (U//FOUO) Gap: Devices must be able to understand and enforce the QoP for a data object while it is in transit.

  (U//FOUO) Recommendations: Enforcement mechanisms must be designed into GIG components that can recognize the QoP parameters and provide the appropriate enforcements. Research must be started immediately to lead to the development of automated solutions, end-to-end QoP enforcement, and standardization of those solutions, to support the GIG 2020 Vision.

3.2.6 (U//FOUO) Protection of High Risk Link Technologies

(U//FOUO) Within the Black Core, packets are protected at the network layer. Network layer protection inherently has traffic analysis, network mapping, and covert channel issues. The risk varies on a link-by-link basis across the Black Core as each link can be characterized as high, medium, or low risk. The definition of these links can be found in The Configuration Guidance for HAIPE Protected Networks, version 2.0.

- (U//FOUO) Gap: Within the Black Core certain links traverse high-risk environments with higher threat of traffic analysis, network mapping, and exfiltration. Cost-effective solutions are required to protect individual links that are characterized as high risk.

  (U//FOUO) Recommendations: Develop a strategy for developing a low-cost protection capability that can be deployed to protect high-risk links. The solution must protect against traffic analysis, network mapping, and prevent an exfiltration path across the link. Solutions must also be easily manageable so that they are not impracticable or are prohibitively costly to use.

3.3 (U) ASSURED ENTERPRISE MANAGEMENT AND CONTROL SUMMARY

(U//FOUO) The technologies that support this cornerstone are organized into seven general categories: Identity Management, Privilege Management, Key Management, Certificate Management, Configuration Management, Policy Management, and Audit Management.

(U) Figure 3.3-1 provides an overview of the technologies and how they support the IA imperatives listed in the Transition Strategy. While none of the technologies will be completed in time to meet the 2008 IA imperatives, the milestones are achievable if current efforts are accelerated.

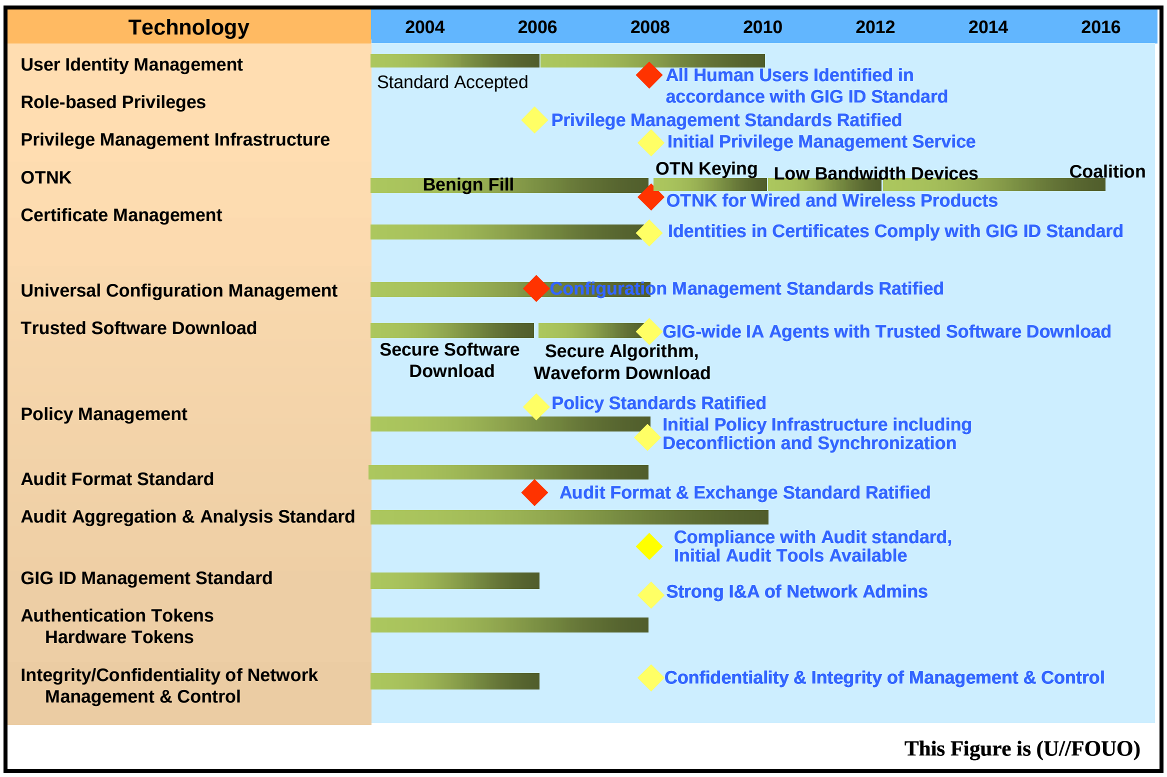


Figure 3.3-1: (U//FOUO) Technology Timeline for Assure Enterprise Management and Control

3.3.1 (U) Identity Management Technologies

(U//FOUO) To meet the 2008 Transition Strategy, a GIG-wide Identity Management standard must be created to describe users, their properties, and profiles. Identity management technologies and standards have traditionally varied dramatically from enterprise to enterprise. For the GIG, a standard should be developed that provides unique, persistent, nonforgeable identities. Since being able to uniquely identify each user is perhaps the most relied on capability in the GIG, developing this standard needs to start immediately.

(U//FOUO) While the 2008 Transition Strategy only requires that Identities be defined for users, to support the 2012 Transition Strategy, Identities also need to be defined for GIG devices and services. Development of an Identity standard for devices and services also needs to start.

- (U//FOUO) Gap: Lack of a unique, persistent, nonforgeable GIG-wide Identity for human users.

  (U//FOUO) Recommendation: Immediately begin development of a GIG Identity standard for human users.

- (U//FOUO) Gap: Lack of a unique, persistent, nonforgeable GIG-wide standard for devices and services.

  (U//FOUO) Recommendation: Begin development of a GIG Identity standard for devices and services.

3.3.2 (U) Inventory Management Technologies

(U) The commercial world is embracing RFID as a technology for inventory management. Although the technology has been available since 1974, standardized, interoperable tags and readers are a recent innovation. Although the field is rapidly developing on its own, the focus is on developing low-cost, passive RFID tags. Consumer privacy issues have raised the concern that RFID tags are still active after leaving the retail sales point and could be used to track individuals, so some work is being done to develop RFID tags that can be killed—rendered permanently inert or unresponsive to a reader interrogation, or rendered temporarily inert. However, no apparent work is being done to develop secure RFID tags which would respond only to interrogation and commands from an authenticated reader.

- (U) Gap: Lack of security for RFID. Lack of ability for RFID tags to only respond to commands from an authorized reader. Lack of an ability to disable RFID tags so they may not be used as tracking devices.

  (U//FOUO) Recommendation: Develop a security architecture and security mechanisms for RFID tags.

- (U) Gap: Lack of a GIG Inventory Management Infrastructure.

  (U//FOUO) Recommendation: Develop a Inventory Management Infrastructure that tracks and manages GIG assets.

3.3.3 (U) Privilege Management Technologies

(U///FOUO) There is an existing standard for privileges (Attribute Certificates) that stems from an extension of the X.509 standard and has been adopted widely by the Public Key Infrastructure (PKI). Attribute Certificates effectively bind privileges to a certificate. Other privilege management approaches also exist that address role-based privileges.

- (U//FOUO) Gap: Lack of definition of privileges necessary to support the GIG.

  (U//FOUO)) Recommendations: Research should be initiated that defines the necessary privilege set and privileges for the GIG. How privileges are stored, retrieved, and managed within the GIG must also be defined so it is scaleable to the GIG enterprise. Rule-based privileges need to be defined for human users, devices, services, and COIs. Role-based privileges also need to be defined so that a GIG entity can dynamically switch between roles and still receive the appropriate privileges. Privileges also need to be defined in the context of the RAdAC model.

- (U//FOUO) Gap: Lack of sufficient support for privileges, trust anchors, and other access control information required by the GIG.

  (U) Recommendations:

  1) (U//FOUO) Develop GIG requirements for privileges, trust anchors, and other access control information required by the GIG. Devise an efficient and scaleable approach and supporting standard for managing this information.

  2) (U//FOUO) Evaluate exiting privilege technologies for meeting the GIG Privilege Management requirements.

(U//FOUO) To meet the 2008 Transition Strategy, the above issues need to be standardized, and initial products conforming to the standards be made available to provide an initial privilege management infrastructure. It is recommended that the standardization activity begin immediately in order to meet this timeline.

3.3.4 (U) Key Management Technologies

(U//FOUO) Some of the technologies in Key Management, such as generation, initial key load and rekeying are quite mature and have been adopted under various classified (e.g., Electronic Key Management System [EKMS]) and unclassified (e.g., DoD Public Key Infrastructure [PKI]) infrastructures. However, the management and distribution of crypto-material still remains a very manually intensive process in some cases. Technologies that reduce the distribution burden, such as Over the Air Distribution (OTAD), are available on a relatively small number of devices. The future Over the Network Keying (OTNK) initiative is expected to further reduce the management burden of key material. Many of the issues that surround technological issues of high assurance with key management practices are being addressed by the Key Management Infrastructure (KMI) initiative.

- (U//FOUO) Gap: Weakness or non-existence of associating policy controls (including dynamic policy changes) in an automated fashion to various aspects of the key management cycle.

  (U//FOUO) Recommendation: Develop standards so that automation can be built into promulgating dynamic policy changes into the necessary rules and regulations so that key registration, packaging, distribution, re-keying, revocation, and destruction work seamlessly and in an up-to-date, situational, manner.

- (U//FOUO) Gap: Lack of sufficient automated key distribution and management techniques.

  Recommendations: Continue to develop the OTNK infrastructure to provide automated key distribution. Either revise OTNK or develop additional automated procedures to meet the needs for tactical and special needs users.

- (U//FOUO) Gap: Lack of standards for unified key labeling, packaging, and distribution formats.

  (U//FOUO) Recommendation: Develop standards to unify key packaging and distribution in order to eliminate or reduce manual error-prone and human access vulnerabilities towards threats. Standards and technologies should include the incorporation of Multi-Level systems and data stores.

- (U//FOUO) Gap: Lack of EKMS support for symmetric and Type 3 keys.

  (U//FOUO) Recommendation: The management of symmetric keys and Type 3 keys needs to be included in the evolution of the KMI.

3.3.5 (U) Certificate Management Technologies

(U//FOUO) The only existing Certificate Management standard is found in the PKI arena. Although PKI has been around for many years, PKI has interoperability limitations at the application and component levels. There are no identified interoperability standards or technologies that specify the interfaces for certificate and data exchange between certificate authorities. There are secure transports currently in use for certificates, but as such, there is no GIG-wide enterprise policy that governs what these access control restrictions should be.

(U//FOUO) There currently is ongoing work to enhance certificate attributes that aim to capture additional significant information such as subject privileges, trust anchor information, and other necessary identity, trust, distribution, and access control information. There are also initiatives that are attempting to specify and collate various levels and classifications of certificates such as the Class 3, Class 4, and Class 5 Government and commercial-type certificates. These efforts should continue since they are necessary for the GIG.

- (U//FOUO) Gap: Lack of definition and infrastructure support for Class 5 certificates.

  (U//FOUO) Recommendations: Develop a standard and the necessary infrastructure for Class 5 certificates.

- (U//FOUO) Gap: Lack of support for the GIG Identity standard in the PKI.

  (U//FOUO) Recommendation: Once the GIG Identity management standard has been approved, PKI must evolve to support the newly defined identities.

- (U//FOUO) Gap: Lack of cryptographic binding of a user/entity’s information and attributes of its public key material and the associated trust anchor. The binding is needed to certify that the private key corresponding to the public key in the certificate is held by the same user/entity. There is a need to have binding strength increase with the strength of the cryptographic algorithm and key length used.

  (U//FOUO) Recommendation: Develop a standard to address the appropriate cryptographic binding of attributes to GIG entity.

3.3.6 (U) Configuration Management Technologies

(U//FOUO) Individual point solutions for various parts of configuration management are mature. There are examples of successfully deployed products in commercial environments. However, none of the technologies meets GIG requirements for the high assurance required to securely manage Information Assurance assets across a lower assurance network.

- (U//FOUO) Gap: Lack of product support for:
    - (U//FOUO) Protected communication paths between configuration management server and managed device
    - (U//FOUO) Authentication of client machines and authentication of configuration management servers
    - (U//FOUO) Ability to model configuration changes before deployment
    - (U//FOUO) Testing configuration. Many products support test deployments before a patch or upgrade deployment, but it is not industry wide
    - (U//FOUO) Authenticated or cryptographic verification of configurations. Current products assumed the device configuration information could be trusted
    - (U//FOUO) Sensitive material distribution, such as keys, which require protection, receipts, and auditable tracking of delivery
    - (U//FOUO) Remote update of firmware

    (U//FOUO) Recommendations: Develop interoperable solutions to the above list of configuration management product gaps to support deployment of GIG-wide Configuration Management agents.

- (U//FOUO) Gap: Lack of Trusted download capability for software, algorithms, and waveforms.

    (U//FOUO) Recommendation: Continue the development of a trusted software download capability.

3.3.7 (U) Policy Management Technologies

(U//FOUO) There are several vendor-specific policy management products available today, but they do not incorporate security attributes required by the GIG into their products. In order to meet the 2008 Transition Strategy standard for policy definition, deconfliction and synchronization need to be developed and ratified. Initial products complying with these standards are also required in order to begin deploying a policy management infrastructure.

- (U//FOUO) Gap: Lack of standards for specifying policy. The policy language needs to cover all GIG policies: access control, quality of protection, quality of service, transport, audit, computer network defense, and policies covering the hardware and software associated with GIG assets.

    (U//FOUO) Recommendations: There are several initiatives to define policy languages. These initiatives must be examined to determine their suitability for the GIG. Security attributes must be inserted into the appropriate policy languages to ensure that the GIG IA policy can be managed.

- (U//FOUO) Gap: Lack of research in policy deconfliction. Existing research has shown that there can be different types of conflicts between policies. In cases where one policy requires a particular action and an overlapping policy does not address that action, the conflicts can generally be resolved. However, when one policy requires an action and an overlapping policy explicitly prohibits it, the conflict can not be resolved by an automated system.

- (U//FOUO) Recommendation: Build on existing research in policy deconfliction to establish general rules and procedures, to automate the deconfliction process to the maximum extent possible.

- (U//FOUO) Gap: Lack of tools that provide policy deconfliction. Current tools require human intervention for policy deconfliction.

  (U//FOUO) Recommendations: Develop technology standards for how to handle IA policy conflicts.

- (U//FOUO) Gap: Lack of standard approaches (push/pull) for policy distribution, including protection of policy at rest and in transit, policy validation, distribution error, and exception handling.

  (U//FOUO) Recommendations: Develop standard approaches to provide policy distribution. Both push and pull policy distribution will be used in the GIG. Standard approaches need to address multiple policy distribution techniques. Develop standards for policy validation, error and exception handling.

- (U//FOUO) Gap: Lack of methods for performing policy synchronization. It is not feasible to assume that policy changes will be implemented instantaneously across the GIG or even across an enterprise. Methods and procedures must be in place to allow a policy to be propagated at a reasonable pace across multiple components.

  (U//FOUO) Recommendations: Develop standards that allow policy changes to be propagated at a reasonable rate across an enterprise. Policy propagation should not create a window of vulnerability during the transition and should not create a denial of service condition.

- (U//FOUO) Gap: Lack of tools for analyzing the affects of policy and multiple policy objects on the GIG. New policy can create undesired conditions through incorrect policy translation or incorrectly formulated policy.

  (U//FOUO) Recommendations: Develop tools for modeling new policy on multiple classes of objects and testing their implementation to verify that policy is being enforced as intended and the new policy is performing the desired changes.

- (U) Gap: Lack of a consistent user interface for managing policy on multiple classes of assets.

  (U//FOUO) Recommendations: Develop tools that can manage multiple classes of assets, including devices from multiple vendors.

- (U//FOUO) Gap: Lack of tools for translating natural language policies into policy base logic.

    (U//FOUO) Recommendations: Develop tools that translate a human understandable (i.e., natural language) policy statement into a configuration file that can be used by a device in the GIG. Translation must also be done in reverse; that is, from a configuration file into natural language policy statements.

3.3.8 (U) Audit Management Technologies

(U//FOUO) Audit Management today exhibits a lack of maturity in standards-based solutions. There are vendors that provide some type of audit capability built into their proprietary solutions. Without standards in technology, interoperability (within components, log formats, audit analyses, etc,), and policies, there is a gap in the unified audit management scheme required by the GIG enterprise.

- (U//FOUO) Gap: Lack of standards in technology, interfaces, interoperability, and policies needed to support Audit Management for the 2008 Transition Strategy.

    (U//FOUO) Recommendations: To ensure that 2008 Transition Strategy for Audit Management can be met, work to define the following set of standards must begin immediately and be completed by 2006:

    1) (U//FOUO) Standard Log and event formats to capture and record normalized GIG-wide activities and system performance

    2) (U//FOUO) Standards for correlation, analysis, and alerting services that subscribe to audit data publishing

    3) (U//FOUO) Standardized Securing of audit data into one-way stores

    4) (U//FOUO) Standardizing Agents and Agentless components for interoperability and security assurance

    5) (U//FOUO) Standards for tools that monitor system resources

    6) (U//FOUO) Central monitoring and interfacing standards

    7) (U//FOUO) Policies on what events are to be audited under what circumstances. This must include: (a) what actions to take when the audit log becomes full (e.g., stop auditing new events; overwrite the oldest existing records; shut down the system); (b) whether auditing can change in an automated manner in response to system events (e.g., if processing load becomes too high, scale back auditing to allocate more resources to production work); (c) what privileges are required to change audit parameters; and (d) deletion of audit records.

- (U//FOUO) Gap: Lack of Audit analysis tools.

    (U//FOUO) Recommendations: Develop Audit Management products and tools that comply with the above list of standards.

3.3.9 (U) Confidentiality & Integrity of Network Management & Control Technologies

(U//FOUO) Solutions currently exist for providing confidentiality and integrity of network management flows. However, they are not widely deployed across the GIG. An effort by GIG programs must be made to provide secure network management solutions.

- (U//FOUO) Gap: Lack of interoperable solutions for providing confidentiality and integrity of network control flows. When solutions exist they are usually specific for a particular protocol. Implementing several protocol unique solutions can impose a heavy management burden on a system without much benefit due to incomplete security solutions.

  (U//FOUO) Recommendations: An approach to providing confidentiality and integrity for all network control protocols needs to be defined such that a secure solution is provided that does not unnecessarily burden the operation of the GIG. Research should be started to develop and socialize this solution with the GIG and commercial industry. Any potential solution must be embraced by commercial industry to have the interoperable implementations required by the GIG.

3.4 (U) CYBER SITUATIONAL AWARENESS AND NETWORK DEFENSE SUMMARY

(U//FOUO) Technologies that support this cornerstone are organized into five general categories: Protection, Monitoring, Detection, Analysis, and Response.

(U) Figure 3.4-1 provides an overview of the technologies and how they support the IA imperatives listed in the Transition Strategy. As shown, while none of the technologies will be completed in time to meet the 2008 IA imperatives, the milestones are achievable if current efforts are accelerated. Some imperatives have no supporting technologies identified in this release of the document. These are discussed in the gaps below.

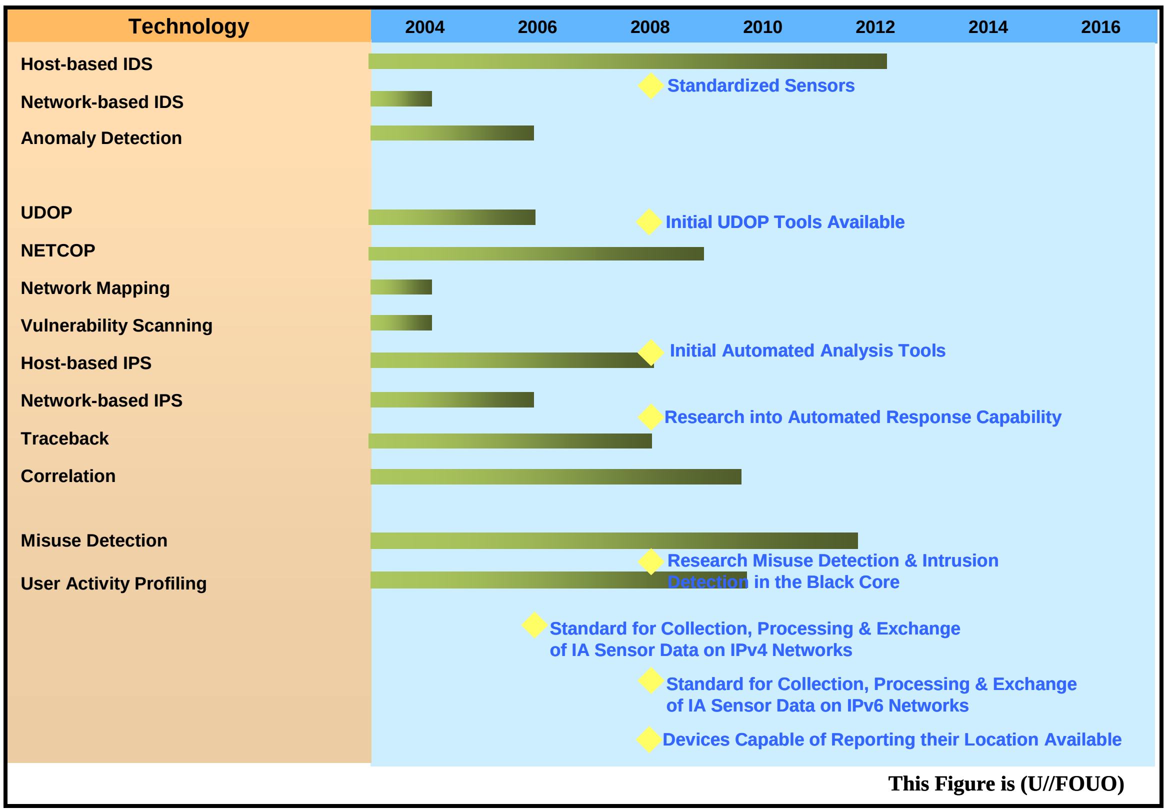


Figure 3.4-1: (U//FOUO) Technology Timeline for Cyber Situational Awareness and Network Defense

3.4.1 (U) Protection Technologies

(U//FOUO) Protection technologies relevant to the GIG have been developed by the commercial market and are—for the most part—mature. However, additional work is needed to adapt these solutions in order to meet the unique requirements imposed by the GIG's environments.

- (U//FOUO) Gap: Protection technologies are static in their operation and require manual configuration.

  (U//FOUO) Recommendation: Research and develop dynamic protection mechanisms capable of modifying device settings (e.g., ports and protocols on network and host-based firewalls) according to current network conditions and published Information Operations Condition (INFOCON).

- (U//FOUO) Gap: Honeynet and Honeypot technologies are well developed, but current implementations do not provide semi-automated operation or support the data volume needed for useful implementation in the GIG.

  (U//FOUO) Recommendation: Existing tools used to capture, control, and analyze data must be enhanced to support automatic filtering of larger amounts of data, correlate with other network operations information (e.g., Intrusion Detection System (IDS) activity), and provide a unified view of ongoing attacks.

3.4.2 (U) Monitoring Technologies

(U//FOUO) State-of-practice approaches for Computer Network Defense (CND) monitoring rely on access to unencrypted traffic and often employ small numbers of sensors located on high-speed backbones. Such approaches are incompatible with the GIG. As the Black Core evolves, unencrypted traffic will be limited—eventually eliminated, so network monitoring must adapt. Moreover, the sheer size and complexity of the GIG mandates use of a large and distributed network of sensors located on lower data bandwidth links. These sensors will provide information to a central correlation function to provide an integrated picture of GIG network state.

(U) Specific gaps and recommendations for monitoring technologies are listed below:

- (U//FOUO) Gap: Current sensor data collection, manipulation, and storage capabilities are insufficient to support the GIG's aggregate bandwidth.

  (U//FOUO) Recommendation: Inexpensive data collection technologies that work at line speed must be developed, along with a scaleable architecture to employ those technologies and sensors.

- (U//FOUO) Gap: Architecture for a sensor grid, capable of supporting the real time data needs of the User Defined Operational Picture (UDOP), has not been defined. This grid would require centralized data storage capabilities sufficient to meet GIG needs.

  (U//FOUO) Recommendation: Devise a technology development plan for deploying a distributed sensor grid across the GIG. The sensor grid must report collected data to a central location. Data storage requirements for the centralized store must be defined.

- (U//FOUO) Gap: The lack of standards for sensor data prevents the design and implementation of a global sensor grid.

  (U//FOUO) Recommendation: Develop a standard for the collection of sensor data. The standard must be integrated into new products to support the 2008 Transition Strategy.

- (U//FOUO) Gap: Current sensors are unable to monitor encrypted packets in the Black Core.

  (U//FOUO) Recommendation: Conduct research aimed at providing CND monitoring capabilities on encrypted traffic in the Black Core.

- (U//FOUO) Gap: Lack of monitoring capabilities for Internet Protocol version 6 (IPv6) networks.

  (U//FOUO) Recommendation: Conduct research aimed at defining monitoring capabilities for IPv6 networks. Incorporate IPv6 monitoring capabilities into GIG situational awareness and monitoring capabilities.

- (U//FOUO) Gap: Current network mapping and discovery tools do not provide the collaborative capabilities across a hierarchical architecture that will be needed to support sophisticated monitoring and controls across the large, complex, and dynamic GIG.

  (U//FOUO) Recommendation: An agent-based collaborative network discovery architecture must be devised and the associated tool technologies developed.

3.4.3 (U) Detection Technologies

(U//FOUO) Detection technologies consist of Intrusion Detection Systems (IDS), Intrusion Protection Systems (IPS), and user profiling. IDSs have been developed and used for years in the commercial market to detect network intrusions and attacks. However, they have not been used on such an expansive network as the GIG. IPSs are a relatively new technology, combining detection and response capabilities in one package. As a detection capability, however, they offer nothing new relative to IDSs. User-profiling is used to detect insider misuse, but these technologies are fairly new and immature.

(U) Specific gaps and recommendations for detection technologies are listed below:

- (U//FOUO) Gap: The current Network-based Intrusion Detection System (NIDS) architectures used by Defense Information Systems Agency (DISA) and the Services are not compatible with the Black Core concept and may not scale well.

  (U//FOUO) Recommendation: Initiate architecture, technology, and standards development efforts to integrate NIDS and Host-based Intrusion Detection System (HIDS) into the global and tiered architecture envisioned for the GIG.

- (U//FOUO) Gap: Anomaly detection offers several significant potential benefits, most notably the ability to detect zero-day attacks. However, current implementations are plagued with high, false-alarm rates that make this technology unusable for GIG applications.

(U//FOUO) Recommendation: Accelerate and guide ongoing research to develop robust anomaly detection capabilities with low false-alarm rates.

- (U//FOUO) Gap: Current intrusion detection capabilities rely on unencrypted packet headers and payloads to detect anomalous activity.

  (U//FOUO) Recommendation: Initiate research to develop advanced intrusion detection systems capable of interoperating on encrypted segments.

- (U//FOUO) Gap: Standards for IDS data and communication are needed to implement a comprehensive and distributed intrusion detection scheme for the GIG. An IETF working group, the Intrusion Detection Working Group (IDWG), is developing standards that will formalize data formats and exchange processes, but this work is not yet complete.

  (U//FOUO) Recommendation: Through participation on the IDWG, the DoD should influence the IDS standards currently under development to meet GIG needs.

- (U//FOUO) Gap: The IDS data exchange requirements for a global network, such as the GIG, could present an undue bandwidth burden on the network being protected. It is not known what these data bandwidth requirements are or what is the best approach for architecting, connecting, and controlling the IDSs across the GIG.

  (U//FOUO) Recommendation: Devise and study architecture alternatives for integrating and controlling GIG IDSs. This study should include trade-offs of key characteristics, such as expected performance, complexity, operational costs (in terms of manpower), and inter-IDS communications bandwidth.

- (U//FOUO) Gap: Detecting insider misuse must rely heavily on user profiling of expected normal behavior as well as on application-specific rules. However, there are significant limitations to this approach, including detectability of slow profile changes and high false alarm rates. Some Government Off-The-Shelf (GOTS) and Commercial Off-The-Shelf (COTS) user profiling tools are available, but much more work is needed to bring their capabilities to maturity.

  (U//FOUO) Recommendation: Development work should be undertaken to determine additional user observables (e.g., websites frequently visited and other individuals with whom the user exchanges e-mail) and to refine existing tools to incorporate this additional information.

3.4.4 (U) Analysis Technologies

(U//FOUO) The breadth, depth, and dynamic nature of the GIG present huge challenges for the analysis component of CND. Correlation processes will have to deal with a large, distributed sensor grid that generates enormous amounts of data from a variety of sources. The manual attack attribution techniques currently used will be inadequate for the expected large volume of network traffic on the GIG.

(U) Specific gaps and recommendations for analysis technologies are listed below:

- (U//FOUO) Gap: There are several different trace-back techniques that have been used with varying success to identify the source of an attack. However, they often feature manual operation and operate on unencrypted packets. These are serious limitations for the GIG.

  (U//FOUO) Recommendation: Continue research and development to advance current techniques for use in the Black Core.

- (U//FOUO) Gap: Existing trace-back approaches are based on correlating similar transactions along a connection path. Many attacks use remote hosts to launch attacks, which are effective at circumventing these trace-back techniques.

  (U//FOUO) Recommendation: Research new techniques that deduce correlation of intent along connection paths, perhaps through a combination of transaction correlation and signature analysis of packet content. Such techniques would be cognizant of attack strategies (i.e., attacks launched through one or more intermediaries) and look for correlations of the resulting packet sequences among hosts across the networks.

- (U//FOUO) Gap: Vulnerability analysis tools consider individual vulnerabilities independent of one another and in the context of a single host. The vulnerability of an enclave or network, however, is determined—in part—by the aggregation of host vulnerabilities. Current tools do not determine aggregate vulnerability.

  (U//FOUO) Recommendation: Extend the capabilities of current vulnerability analysis tools to include Topological Vulnerability Analysis across groups of hosts in networks.

- (U//FOUO) Gap: The large number of sensors disbursed across multiple hierarchical levels of the GIG represents huge challenges for analysis. Correlation technology must be able to handle large volumes of intrusion detection data in real time, fuse heterogeneous data from disparate levels in the global network hierarchy, and accommodate other operational factors, such as typical adversary behavior, normal network activity, and mission critical components and applications. No technologies exist to provide this analysis capability for such a large network.

  (U//FOUO) Recommendation: Research must be undertaken to devise a unified correlation and analysis approach for the GIG CND effort. In addition to correlation of intrusion detection data, focus should include key performance measures critical for a GIG-sized network, such as dropped-alert rate, false alarm rate, bandwidth for communication between distributed processing nodes, and processing latency. Part of this work must be an analysis-of-alternatives (AoA) study to determine sensor grid architecture limitations imposed by the analysis approach.

3.4.5 (U) Response Technologies

(U//FOUO) Today, responses to computer network attacks are largely manual, because available tools are limited in their capabilities, and uncertainties exist on the impact of automated responses on the mission of the enterprise. However, due to its size and criticality, the GIG will require an automated or at least semi-automated response to network attacks.

(U) Specific gaps and recommendations for response technologies are listed below:

- (U//FOUO) Gap: CND analysts and warfighters must understand, a priori, the operational implications of shutting down or restricting capabilities in response to network attack. Until combatants are able to fully understand the implications of network response actions to attacks, automated response capabilities will not be adopted. Some research has been done in developing tools for assessing operational impact of attack responses. However, these tools have not evolved to the point where they can provide the user an estimate of impact on specific missions.

    (U//FOUO) Recommendation: Accelerate and guide research for modeling the impact of attack response on warfighting operations in context of the GIG.

- (U//FOUO) Gap: A semi-automated approach for responding to attacks is needed. A fully manual process permits deliberate consideration of response options and more complete attack analysis, but it is labor intensive and takes more time (especially for the GIG), so attack damage could be more widespread. Automated responses can quickly contain an attack, but the impact on the network can be unpredictable and unnecessarily restrict warfighting operations. For maximum response effectiveness in the GIG, a balance between manual and automated response is needed, but no work has yet been done to determine how this could be done.

    (U//FOUO) Recommendation: Continue research to determine the best approach for a semi-automated response to network attack, taking into consideration effectiveness of attribution activities, impact of response on warfighting operations, and manpower required.

- (U//FOUO) Gap: Current response capabilities are limited to simplistic point solutions, such as blocking a port and IP address pair at the network boundary, which will become ineffective against sophisticated attacks.

(U//FOUO) Recommendation: Continue research into sophisticated response capabilities applied to distributed network components.

4 (U) ACRONYMS AND ABBREVIATIONS

AA	Attribute Authority
ABAC	Attribute-Based Access Control
ABNF	Augmented Backus-Naur Format
AC	Attribute Certificate Access Control
ACAP	Application Configuration Access Protocol
ACE	Advanced Computing Environment
ACL	Access Control List
ACOA	Alternate Course Of Action
ACP	Allied Communications Publication
ACRL	Attribute Certificate Revocation List
ACS	Access Control Server
AEHF	Advanced Extremely High Frequency
AES	Advanced Encryption Standard
AFS	Agent Functional Stack
AH	Authentication Header
AIC	Adaptive Information Control
AICE	Agile Information Control Environment
A/J	Anti-Jam
AKA	Authentication and Key Agreement
a.k.a	Also known as
AKP	Advanced Key Processor
ANDVT	Advanced Narrowband Digital Voice Terminal
ANSI	American National Standards Institute

AODV	Ad-hoc On-Demand Distant Vector
API	Application Programming Interface
AS	Autonomous System
ASD	Assistant Secretary of Defense
ASIC	Application-Specific Integrated Circuit
ASN1	Abstract Syntax Notation One
AS&W	Attack, Sensing & Warning
ASCII	American Standard Code for Information Interchange
ASN.1	Abstract Syntax Notation
ATM	Asynchronous Transfer Mode
AVLAN	Authenticated Virtual Local Area Network
BAAD	Battlefield Awareness and Data Dissemination
BC	Biometric Consortium
BEM	Biometric Evaluation Methodology
BER	Basic Encoding Rules Bit Error Rate
BET	Bulk Encrypted Transaction
BGP	Border Gateway Protocol
BIOS	Basic Input-Output System
BIS	Boot Integrity Services
BMO	Biometric Management Office
BOOTP	Boot Protocol
BoSS	Baystack Operating System Switching Software
BSP	Biometric Service Providers
C2	Command and Control

C2G	Command and Control Guard
C3I	Command, Control, Communications and Intelligence
CA	Certification Authority
CAC	Common Access Card
CAPCO	Controlled Access Program Coordinator Office
CAPI	Crypto API
CAW	Certification Authority Workstation
CBC	Cipher Block Chaining
CBEFF	Common Biometric Exchange Formats Framework
CBIS	Content-Based Information Security
CC	Common Criteria
CCITT	Consultative Committee on International Telegraphy and Telephony
CDMA	Code Division Multiple Access
CDS	Cross-Domain Solutions
CDSA	Common Data Security Architecture
CEM	Common Evaluation Methodology Constructive Key Management
CENTRIXS	Combined Enterprise Regional Information Exchange System
CEP	Certificate Enrollment Protocol
CER	Canonical Encoding Rules
CERIAS	Center for Education and Research in Information Assurance and Security
CERT	Computer Emergency Readiness Team
CES	Core Enterprise Service
CHAP	Challenge Handshake Authentication Protocol
CIDF	Common Intrusion Detection Framework

CIF	Component Impact Factor
CIFS	Common Internet File System
CIM	Common Information Model
CJCSI	Chairman of the Joint Chiefs of Staff Instruction
CKL	Compromised Key List
CKM	Constructive Key Management
CLF	Common Log Format
CLI	Command Line Interface
CM	Configuration Management
CMC	COMSEC Material Control
CMCS	COMSEC Material Control System
CMI	Certificate Management Infrastructure
CMMF	Certificate Management Message Format
CMP	Certificate Management Protocol
CMS	Cryptographic Message Syntax
CND	Computer Network Defense
COA	Course of Action
COI	Community of Interest
COMPUSEC	Computer Security
COMSEC	Communications Security
CONOP	Concept of Operation
CONUS	Continental United States
COP	Common Operating Picture
CoP	Coalition Partner
COPS	Common Open Policy Service

CORBA	Common Object Request Broker Architecture
CoS	Class of Service
COTS	Commercial-off-the-Shelf
COWANS	Coalition Operational Wide Area Networks
CPS	Certification Practice Statement
CPU	Central Processing Unit
CRD	Capstone Requirements Document
CRL	Certificate Revocation List
CRMF	Certificate Request Message Format
CSEE	Computer Science and Electrical Engineering
CSIRT	Computer Security Incident Response Team
CSP	Common Security Protocol
CSRC	Contributing Source Real-time Content
CVE	Common Vulnerabilities and Exposures
DAC	Discretionary Access Control
DACP	Digital Access Control Policy
DAML	DARPA Agent Markup Language
DARPA	Defense Advanced Research Projects Agency
DAV	Distributed Authoring & Versioning
DBMS	Database Management System
DCID	Director of Central Intelligence Directive
DCIS	Defense Cross-credentialing Identification System
DDDS	Dynamic Delegation Discovery System
DDES	Double Data Encryption Standard
DDMS	DoD's Discovery Metadata Specification

DDoS	Distributed Denial of Service
DEERS	Defense Enrollment Eligibility Reporting System
DEFCON	Defense Condition
DER	Distinguished Encoding Rules
DES	Data Encryption Standard
DeSiDeRaTa	Dynamic Scalable Dependable Real-Time systems
DH-CHAP	Diffie-Hellman augmented CHAP
DHCP	Dynamic Host Control Protocol
DHS	Department of Homeland Security
DIA	Defense Intelligence Agency
DIACAP	DoD Information Assurance Policy for IA Certification and Accreditation
DII	Defense Information Infrastructure
DIO	Defensive Information Operations
DISA	Defense Information Systems Agency
DISN	Defense Information Systems Network
DITSCAP	DoD Information Technology Security Certification and Accreditation Process
DMDC	Defense Manpower Data Center
DME	Distributed Management Environment
DMI	Desktop Management Interface
DMS	Defense Message System
DMTF	Distributed Management Task Force
DN	Distinguished Name
DNS	Domain Name System
DoD	Department of Defense

DoDI	Department of Defense Instruction
DoS	Denial of Service
DPM	Digital Policy Management
DR/COOP	Disaster Recovery and Continuous Operations
DSA	Digital Signature Algorithm
DSS	Digital Signature Service
DTD	Document Type Definition Data Transfer Device
dBµA	decibels, micro-amps per meter
EAL	Evaluation Assurance Level
EAN.UCC	European Article Number, Uniform Code Council
EAP	Extensible Authentication Protocol
EAS	Electronic Article Surveillance
EIAU	End Information Assurance Unit
ECC	Elliptic Curve Cryptography Error Correcting Code
ECDSA	Elliptic Curve Digital Signature Algorithm
ECU	End Cryptographic Unit
EIAU	End Information Assurance Unit
EIGRP	Enhanced Interior Gateway Routing Protocol
EKMS	Electronic Key Management System
ELF	Extended Log Format
eMASS	Enterprise Mission Assurance Support System
EMSEC	Emission Security
ENUM	Electronic Numbering
EOTN	Encrypted Optical Transport Network

EPC	Electronic Product Code
ESG	Enterprise Wide Sensor Grid
ESM/NM	Enterprise Service Management/Network Management
ESP	Encapsulating Security Payload
ESS	Enhanced Security Services
ETSI	European Technical Standards Institute
FAQ	Frequently Asked Questions
FAR	False Acceptance Rate
FC	Fibre Channel
FCAPS	Fault, Configuration, Accounting, Performance, and Security
FC-GS-3	Fibre Channel–Generic Services–3
FCIP	Fibre Channel over TCP/IP (RFC 3821)
FCsec	Fibre Channel Security
FC-SP	Fibre Channel–Security Protocol
FFRDC	Federally Funded Research and Development Center
FIPS	Federal Information Processing Standards
FIRE	Flexible Intra-AS Routing Environment
FISMA	Federal Information Security Management Act
FiXs	Federated Identity Cross-credentialing System
FMR	False Match Rate
FNBDT	Future Narrow Band Digital Terminal
FNMR	False Non-Match Rate
FOUO	For Official Use Only
FPKI	Federal Public Key Infrastructure
FRR	False Rejection Rate

FTP	File Transfer Protocol
Gbps	Giga bits per second
GCCS	Global Command and Control System
GCP	Gateway Control Protocol
GDS	Global Directory Services
GES	Global Enterprise Service
GIAI	Global Individual Asset Identifier
GID	General Identifier
GIG	Global Information Grid
GIG-BE	Global Information Grid–Bandwidth Expansion
GIG ES	Global Information Grid Enterprise Services
GLN	Global Location Number
GOTS	Government Off-The-Shelf
GRAI	Global Returnable Asset Identifier
GSM	Global System for Mobile (communication)
GSS	Generic Security Services
GTC	Generic Token Card
GTIN	Global Trade Item Number
GUI	Graphical User Interface
GULS	Generic Upper Layer Security
GW	GateWay
HAIPE	High Assurance Internet Protocol Encryptor
HAIPIS	High Assurance Internet Protocol Interoperability Specification
HAPKI	High Assurance Public Key Infrastructure
HBA	Host Bus Adapter

HI	Horizontal Integration
HIDS	Host-Based Intrusion Detection System
HIPPA	Health Information Protection and Privacy Act
HIPS	Host-Based Intrusion Prevention System
HLS	Home Land Security
HMAC	Keyed-Hashing for Message Authentication
HMAC-MD5	Hashed Message Authentication Code-Message Digest Algorithm 5
HMMA	Hypermedia Management Architecture
HNMP	Hierarchical Network Management Protocol
HNMS	Hierarchical Network Management System
HRS	Human Recognition Services
HSM	Hardware Security Module
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
I&A	Identification and Authentication
IA	Information Assurance
IAA SPO	Information Assurance Architecture Special Program Office
IAC	Information Assurance Component
IAD	Information Assurance Directorate
IANA	Internet Assigned Numbers Authority
IATF	Information Assurance Task Force
IAVA	Information Assurance Vulnerability Alert
IAVM	Information Assurance Vulnerability Management
I&W	Indications and Warnings
IB	In Band

IC	Intelligence Community
ICMP	Internet Control Message Protocol
IDC	International Data Corporation
ICSIS	Intelligence Community System for Information Sharing
IDIP	Intrusion Detection and Isolation Protocol
IDMEF	Intrusion Detection Message Exchange Format
IDS	Intrusion Detection System
IDU	Interface Data Unit
IDUP	Independent Data Unit Protection
IDWG	Intrusion Detection Working Group
IDXP	Intrusion Detection eXchange Protocol
IdM	Identification Management
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IESG	Internet Engineering Steering Group
IETF	Internet Engineering Task Force
IFF	Identification Friend-or-Foe
IHMC	Interdisciplinary Study of Human & Machine Cognition
IIA SPO	Information Assurance Architecture Special Program Office
IKE	Internet Key Exchange
IMAP	Internet Message Access Protocol
INE	In-line Network Encryptor
INFOCON	Information Operations Condition
INFOSEC	Information Security
INDEF	Incident Object Description Exchange Format

IOC	Initial Operational Capability
IP	Internet Protocol
IPM	Information Policy Management
IPS	Intrusion Prevention System
IPsec	Internet Protocol Security (IP Security)
IPSRA	Internet Protocol Security Remote Access
IPT	Integrated Product Team
IPv4	IP version 4
IPv6	IP version 6
IPX	Internetwork Packet Exchange
IRM	Information Resources Management
ISAKMP	Internet Security Association and Key Management Protocol
ISDN	Integrated Services Digital Network
IS-IS	Intermediate System-to-Intermediate System
ISM	Information Security Markings
ISSE	Imagery Support Server Environment
ISO	International Organization for Standardization
ISP	Internet Service Provider
IT	Information Technology
ITSEC	Information Technology Security Evaluation Criteria
ITU	International Telecommunications Union
ITU-T	International Telecommunication Union Telecommunication Standardization Sector
IV	Initialization Vector
iFCP	Internet Fiber Channel Protocol (Internet Draft)

iSCSI	Internet SCSI (RFC 3720)
iSNS	Internet Storage Name Service
JAS	Java Agent Services
JTF	Joint Task Force
JTRS	Joint Tactical Radio System
JV2020	Joint Vision 2020
JWICS	Joint Worldwide Intelligence Communication System
KAoS	Knowledgeable Agent-oriented System
KDC	Key Distribution Center
KEK	Key Encryption Key
KMI	Key Management Infrastructure
KVM	Keyboard, Video, Mouse
KMP	Key Management Policy
KMPS	Key Management Policy Server
KMS	Key Management System
KRSS	Key Registration Service Specification
L2TP	Layer 2 Tunneling Protocol
LAN	Local Area Network
LBAC	List Based Access Control
LCD	Liquid Crystal Display
LDAP	Lightweight Directory Access Protocol
LEAP	Lightweight Extensible Authentication Protocol
LIMFAC	List Limiting Factors
LPD	Low Probability of Detection
LPI	Low Probability of Interception

LRS	Linear Recursive Sequence
LSP	Labeled Switch Path
LTANS	Long-Term Archive and Notary Services
LTS	Laboratory for Telecommunications Science
M&C	Management and Control
MAC	Mandatory Access Control Media Access Control Message Authentication Code Mission Assurance Category
MANET	Mobile Ad hoc Network
MAPKI	Medium Assurance Public Key Infrastructure
MC	Multipoint Controller
MCU	Multipoint Control Unit
MD5	Message Digest Algorithm 5
MEGACO	MEdia GAteway COntrol protocol
MELP	Mixed Excitation Linear Prediction
MER	Minimum Essential Requirement
MG	Media Gateway
MGCP	Media Gateway Control Protocol
MIB	Management Information Base
MID	Message Identifier
MIB	Management Information Base
MILS	Multiple Independent Levels of Security
MIME	Multipurpose Internet Mail Extensions
MISSI	Multilevel Information Systems Security Initiative
MITM	Man in the Middle

MKI	Master Key Identifier
MLPP	Multi-Level Precedence and Preemption
MLS	Multi-Level Secure Multiple Levels of Security
MLTC	Multi-Level Thin Client
MNIS	Multi-National Information Sharing
MOSS	MIME Object Security Services
MP	Multipoint Processor
MPL	Mozilla Public License
MPLS	Multi-Protocol Label Switching
MQV	Menezes-Qu-Vanstone
MR	Modem Relay
MS-CHAP	Microsoft Challenge Handshake Authentication Protocol
MS&MS	Mission Support and Management Systems
MSE	Mobile Subscriber Equipment
MSGFMT	Message Format
MSI	Microsoft Installer
MSL	Multiple Single Levels
MSLS	Multiple Single Levels of Security
MSP	Message Security Protocol Metadata Standard for Publication
MTA	Mail Transfer Agent
MW	milli-Watt
Mbps	Megabits per second
NAC	Network Admission Control
NAS	Network Attached Storage

NASA	National Aeronautics and Space Administration
NATO	North Atlantic Treaty Organization
NCES	Network-Centric Enterprise Services
NCOW	Network-Centric Operations and Warfare
NCOW-RM	Net-Centric Operations and Warfare-Reference Model
NCW	Net-Centric Warfare
NDA	National Distribution Authorities
NDS	Novell Directory Service
NETOPS	Network Operations
NFS	Network File System
NGSCB	Next Generation Secure Computing Base
NIAP	National Information Assurance Partnership
NIDS	Network-Based Intrusion Detection System
NII	Networks and Information Integration
NIPRNet	Non-secure Internet Protocol Router Network
NIPS	Network-Based Intrusion Protection System
NISPOM	National Industrial Security Program Operating Manual
NIST	National Institute of Standards and Technology
NMCC	National Military Command Center
NNIDS	Network Node Intrusion Detection System
NNTP	Network News Transport Protocol
NOC	Network Operations Center
NP	Network Processor
NSA	National Security Agency
NTP	Network Time Protocol

15687	OASIS	Organization for the Advancement of Structured Information Standards
15688	OCSP	Online Certificate Status Protocol
15689	OED	OSIS Evolutionary Development
15690	OEM	Original Equipment Manufacturer
15691	OLAC	Object Level Access Control
15692	ONR	Office of Naval Research
15693	OOB	Out of Band
15694	OPIE	One-time Password in Everything
15695	OPS	Optivity Policy Services
15696	OPSEC	Operations Security
15697	OS	Operating System
15698	OSD	Office Secretary of Defense
15699	OSF	Open Software Foundation
15700	OSI	Open Systems Interconnection
15701	OSIS	Open System Information System
15702	OSPF	Open Shortest Path First
15703	OTAD	Over-The-Air Distribution
15704	OTAR	Over-The-Air Rekey
15705	OTAT	Over-The-Air Transfer
15706	OTNK	Over the Network Keying
15707	OTP	One Time Password
15708	OUSPG	Oulu University Secure Programming Group
15709	OVAL	Open Vulnerability Assessment Language
15710	OWL	Web Ontology Language
15711	PAC	Positive Access Control

PANA	Protocol for carrying Authentication for Network Access
PBR	Policy-Based Routing
PBX	Private Branch Exchange
PC	Personal Computer
PCAP	Packet Capture
PC/SC	Personal Computer/Smart Card
PCI	Protocol Control Information
PDA	Personal Digital Assistant
PDP	Policy Decision Point
PDU	Protocol Data Unit
PEAP	Protected Extensible Authentication Protocol
PEM	Privacy Enhanced Mail
PEP	Policy Enforcement Point
PGP	Pretty Good Privacy
PIC	Pre-IKE Credential Provisioning
PIN	Personal Identification Number
PIP	Policy Input Point
PK	Public Key
PKC	Public Key Certificate
PKCS	Public Key Cryptographic Standard
PKI	Public Key Infrastructure
PKINIT	Public Key Initialization Authentication
PKIX	Pubic Key Infrastructure X.509
PKCS	Public Key Cryptography Standards
PMI	Privilege Management Infrastructure

POC	Point of Contact
POD	Proof of Delivery
POM	Program Objective Memorandum
POP	Point of Presence
POP3	Post Office Protocol
PoS	Priority of Service
POTS	Plain Old Telephone System
PP	Protection Profiles
PPK	Pre-Placed Key
PPP	Point-to-Point Protocol
PRBAC	Partition Rule-Based Access Control
PRSN	Primary Services Node
PSEQN	Payload Sequence Number
PSN	Product Services Node
PST	Provision Service Target
PSTN	Public Switched Telephone Network
PXE	Preboot eXecution Environment
QoP	Quality of Protection
QoS	Quality of Service
RA	Registration Authority Response Action
RAdAC	Risk Adaptable Access Control
RADIUS	Remote Access Dial In User Service
RAID	Recent Advances in Intrusion Detection
RBAC	Role-Based Access Control

RCD	Reference Capability Document
RDEP	Remote Data Exchange Protocol
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
RF	Radio Frequency
RFC	Request for Comments
RFID	Radio Frequency Identification
RFP	Request for Proposal
RIP	Routing Information Protocol
RM	Radiant Mercury
ROI	Return on Investment
RPC	Remote Procedure Call
RPSLng	Routing Policy Specification Language Next Generation
RR	Receiver Report
RREP	Route Reply
RREQ	Route REQuest
RSA	Rivest Shamir Adelman (public key encryption algorithm)
RSVP	ReSerVation Protocol
RTCP	Real Time Control Protocol
RTP	Real Time Protocol
RVN	Red Virtual Network
rDSA	Reversible Public Key Cryptography for Digital Signatures
S/Key	Shared Key
SA	Security Association Situational Awareness

15787	SACRED	Securely Available Credentials
15788	SAD	Security Association Database
15789	SAMI	Source and Method Information
15790	SAML	Security Assertion Markup Language
15791	SAN	Storage Area Network
15792	SAODV	Secure Ad hoc On Demand Distance Vector
15793	SAP	Service Access Point
15794	SAR	Security Aware ad-hoc Routing
15795	SASL	Simple Authentication Security Layer
15796	SBSM	Session-Based Security Model
15797	SBU	Sensitive But Unclassified
15798	SCA	Subordinate Certificate Authority
15799	SCEP	Simple Certificate Enrollment Protocol
15800	SCI	Sensitive Compartmented Information
15801	SCIF	Sensitive Compartmented Information Facility
15802	SCP	Secure Copy
15803	SCSI	Small Computer System Interface
15804	SDNS	Secure Data Network System
15805	SEI	Software Engineering Institute
15806	SEM	Security Event Management
15807	SEQN	Sequence Number
15808	SESA	Symantec Enterprise Security Architecture
15809	SESE	Security Exchange Service Element
15810	SHA	Secure Hash Algorithm
15811	SHF	Super High Frequency

S-HTTP	Secure HTTP
SIF	System Impact Factor
SIGINT	Signals Intelligence
SIM	Subscriber Identity Module
SIP	Session Initiation Protocol
SIPRNet	Secret Internet Protocol Router Network
SISWG	Security in Storage Working Group (IEEE)
SLA	Service Level Agreement
SLP	Service Location Protocol (RFC 2608)
SMBIOS	Systems Management Basic Input/Output System
SMI	Security Management Infrastructure Security Management Information
SMI-S	Storage Management Initiative - Specification
S/MIME	Secure Multipurpose Internet Mail Extensions
SMS	System Management Server
SMTP	Simple Mail Transfer Protocol
SMUX	SNMP Multiplex
SMW	Security Management Workstation
SNIA	Storage Network Industry Association
SNMP	Simple Network Management Protocol
SOA	Source of Authority
SOAP	Simple Object Access Protocol
SOC	Security Operations Center
SoM	Strength of Mechanism
SONET	Synchronous Optical NETwork

SPAWAR	Space and Naval Warfare Systems Command
SPD	Security Policy Database
SPF	Shortest Path First
SPI	Service Provider Interface Security Parameters Index
SPIE	Source Path Isolation Engine
SPIF	Security Policy Information File
SPKI	Simple Public Key Infrastructure
SPKM	Simple Public-Key GSS-API Mechanism
SPML	Services Provisioning Markup Language
SPRT	Simple Packet Relay Transport
SR	Sender Report
SRD	Short Range Device
SRTCP	Secure Real Time Control Protocol
SRTP	Secure Real Time Protocol
SSCC	Serial Shipping Container Code
SSE	State Signaling Events
SSH	Secure Shell
SSL	Secure Session Layer Secure Sockets Layer
SSO	Single Sign-On
SSP	Secure Server Protocol
SSRC	Synchronization Source Real-time Content
SSTC	Security Services Technical Committee
STE	Secure Teleconferencing Equipment
STS	Security Token Service

STU	Secure Telecommunications Unit
SVoIP	Secure Voice over Internet Protocol
SWRL	Semantic Web Rule Language
TA	Traffic Analysis
TAMP	Trust Anchor Management Protocol
TC	Transformational Communications
TCG	Trusted Computing Group
TCM	Transformational Communications MILSATCOM
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TCSEC	Trusted System Evaluation Criteria
TEK	Traffic Encryption Key
TFS	Traffic Flow Security
TFTP	Trivial File Transfer Protocol
TGS	Trusted Gateway Solution
TGT	Ticket Granting Ticket
TLS	Transport Layer Security
TMF	TeleManagement Forum
TN	Traffic Normalizer
TPED	Task, Process, Exploit, and Disseminate
TPM	Trusted Platform Module
TPPU	Task, Post, Process, and Use
TRANSEC	Transmission Security
TRL	Technology Readiness Level
TSAT	Transformational Satellite

TSP	Time-Stamp Protocol
TVA	Topological Vulnerability Analysis
TV-1	Technical View current state
TV-2	Technical View future state
U	Unclassified
UDOP	User Defined Operational Picture
UDP	Universal Datagram Protocol
UMBC	University of Maryland Baltimore County
UMTS	Universal Mobile Telecommunications System
UPN	User Personalized Network
URI	Universal Resource Identifier
USB	Universal Serial Bus
USD/ATL	Undersecretary of Defense for Acquisitions, Technology and Logistics
USM	User-based Security Model
USSTRATCOM	U.S. Strategic Command
UUID	Universal Unique ID
VACM	View-based Access Control Model
VI	Vulnerability Index
VKB	Virtual Knowledge Base
VLAN	Virtual Local Area Network
VM	Virtual Machine
VoIP	Voice over IP
VoSIP	Voice over Secure IP
VPA	Virtual Port-based Authentication
VPN	Virtual Private Network

W3C	World Wide Web Consortium
WAM	Web Access Management
WAN	Wide Area Network
WAP	Wireless Application Protocol
WBEM	Web-Based Enterprise Management
WebDAV	Web Distributed Authoring and Versioning
WebDAV-AC	Web Distributed Authoring and Versioning-Access Control
WG	Working Group
WIN-T	Warfighter Information Network-Tactical
WLAN	Wireless Local Area Network
WNW	Wideband Networking Waveform
WPA	Wi-Fi Protected Access
WS	Web Services
WSDL	Web Services Description Language
WSF	Web Services Framework
WS-I	Web Services Interoperability
WSS	Web Services Security
WS-Trust	Web Services–Trust Language
WTLS	Wireless Transport Layer Security
WXS	W3C XML Schema
XACML	eXtensible Access Control Markup Language
XCMS	XML Cryptographic Message Syntax
XER	XML Encoding Rules
X-KISS	XML Key Information Service Specification
XKMS	XML Key Management Specification

X-KRSS	XML Key Registration Service Specification
XML	Extensible Mark-up Language
XML_DSIG	XML Digital Signature
XML_ENC	XML Encryption
XrML	eXtensible Rights Markup Language

(U) Appendices

(U//FOUO) APPENDIX A: MAPPING OF TECHNOLOGIES TO IA SYSTEM ENABLERS

(U) The following Table lists the detailed technologies explored in this document. Each is mapped to the Technology Category under which it is discussed and the IA System Enabler section where it can be located in the document. The goal is for this to help readers locate the technologies in which they are interested.

Table A-1: (U//FOUO) Mapping of Technologies to IA System Enablers

This Table is (U//FOUO)

Technology Category	Detailed Technology	IA System Enabler
Authentication Tokens	Asynchronous Synchronous Time-driven Event-driven	2.1 Identification and Authentication
Biometrics	Physiological Fingerprint Face Recognition Iris Recognition Hand and Finger Geometry Behavioral Signature Verification Speech Recognition	2.1 Identification and Authentication
Device/Service authentication	Strong Authentication for Devices	2.1 Identification and Authentication
Authentication protocols	802.1x for Network Applications 802.1x for Device Authentication Manufacturing Time Device Credentials Web Service Protocol for Business-Application Integration Application Connectors and Authentication Clients Credential Provisioning and Validation	2.1 Identification and Authentication
Single Sign-On	Early SSO Techniques Scripting Password Synchronization LDAP Directories SSO Architectures Centralized Model Federated Model Kerberos PKI Certificates SAML	2.1 Identification and Authentication

This Table is (U//FOUO)		
Technology Category	**Detailed Technology**	**IA System Enabler**
Authentication Confidence	Authentication Confidence	2.1 Identification and Authentication
Core RAdAC	Core RAdAC	2.2 Policy-Based Access Control
Assured Metadata	Metadata Language & Standards Trusted Metadata Creation Tools Crypto-binding of Metadata to Source Information Objects	2.2 Policy-Based Access Control
Digital Access Control Policy	Digital Access Control Policy	2.2 Policy-Based Access Control
Protecting Data-at-Rest	Cryptography Data Backup & Archive Data Destruction Labeling Periods Processing Physical Controls Quality of Protection	2.3 Protection of User Information
Protecting Data-in-Transit	Application Layer Technologies Non-Real-Time Data Technologies Traditional Application Security Session Security SSL/TLS GULS Web Services Security Real-Time Data Technologies FNBDT Interoperability/Gateways Secure VoIP RTP and RTCP Transport & Network Layer Technologies Non-Real-Time Data Technologies IP Layer Security TFS VPN Real-Time Data Technologies Secure VoIP Call Control Link & Physical Layer Technologies Anti-Jam Link Encryption TRANSEC	2.3 Protection of User Information

This Table is (U//FOUO)		
Technology Category	**Detailed Technology**	**IA System Enabler**
Trusted Platforms	Trusted Platforms	2.3 Protection of User Information
Trusted Applications	Trusted Applications	2.3 Protection of User Information
Cross Domain Solutions	Cross Domain Solutions	2.3 Protection of User Information
Non-Repudiation	Non-Repudiation	2.3 Protection of User Information
Development of Policies	Centralized vs. Distributed Elements of Policies Access Control Trust Anchors Policy Languages	2.4 Dynamic Policy Management
Distribution of Policies	Standard Protocols Security Issues	2.4 Dynamic Policy Management
Policy Management Architectures	Policy Directories	2.4 Dynamic Policy Management
IA Policy-based Routing	IA Policy-based Routing	2.5 Assured Resource Allocation
Operational-based Resource Allocation	Operational-based Resource Allocation	2.5 Assured Resource Allocation
Integrity of Network Fault Monitoring/Recovery and Integrity of Network Management & Control	Integrity of Network Fault Monitoring/Recovery and Integrity of Network Management & Control	2.5 Assured Resource Allocation
Protect Technologies	Protect Technologies	2.6 Network Defense and Situational Awareness
Deception Technologies	Honeypots Honeynets	2.6 Network Defense and Situational Awareness
Situational Awareness	UDOP NETOPS	2.6 Network Defense and Situational Awareness
Network Mapping	Network Mapping	2.6 Network Defense and Situational Awareness
IDS	Host-based IDS Network-based IDS	2.6 Network Defense and Situational Awareness
IPS	IPS	2.6 Network Defense and Situational Awareness
User Activity Profiling	User Activity Profiling	2.6 Network Defense and Situational Awareness
Cyber Attack Attribution	Hop-by-Hop Traceback Backscatter Traceback CenterTrack ICMP Traceback or iTrace Hash-based IP Traceback	2.6 Network Defense and Situational Awareness

This Table is (U//FOUO)		
Technology Category	**Detailed Technology**	**IA System Enabler**
Correlation Techniques	Correlation Techniques	2.6 Network Defense and Situational Awareness
CND Response Actions	CND Response Actions	2.6 Network Defense and Situational Awareness
Automated IAVA Patch Management	Automated IAVA Patch Management	2.6 Network Defense and Situational Awareness
Identity Management	Identity Management	2.7 Management of IA Mechanisms and Assets
Privilege Management	Rules-based Authorization Schemes Roles-based Authorization Schemes PMI	2.7 Management of IA Mechanisms and Assets
Key Management	Evolution of Key-based Equipment Technology KMI XML Key Management Services Constructive Key Management IKE and ISAKMP Hardware Security Module	2.7 Management of IA Mechanisms and Assets
Certificate Management	Certificate Management	2.7 Management of IA Mechanisms and Assets
Configuration Management of IA Devices and Software	Systems Management Applications Network Boot Applications Malware Management ECU Update Patch Management Systems	2.7 Management of IA Mechanisms and Assets
Inventory Management	Inventory Management	2.7 Management of IA Mechanisms and Assets
Compromise Management of IA Devices	Compromise Management of IA Devices	2.7 Management of IA Mechanisms and Assets
Audit Management	Audit Management	2.7 Management of IA Mechanisms and Assets
The Table is (U//FOUO)		

15964

(U//FOUO) APPENDIX B: TV-1 FOR IA

(U) The DoD Architecture Framework (DoDAF) provides a convenient repository for describing some of the content from the Roadmap, but there is no “system” for which a system architecture is being described. From DoDAF: “The TV includes a collection of the technical standards, implementation conventions, standards options, rules, and criteria organized into profile(s) that govern systems and system elements for a given architecture.” The set of standards approved to support the existing capability (as-is) includes those standards listed in the DoD IT Standards Registry (DISR) Baseline Release 04-1.0. Table B-1: Technical Standards Profile identifies standards that apply to systems view elements. The standards in this table are a summary of the standards identified in the Section 3—existing standards identified as needed to satisfy capabilities listed in the GIG IA Reference Capabilities Document (RCD).

Table B-1: (U//FOUO) TV-1 for IA

This Table is (U//FOUO)

Name	Description
PKCS #11	Cryptographic Token Interface (cryptoki) Standard (specification of an application programming interface API for cryptographic token devices)
PKCS #12	Personal Information Exchange Syntax (specifies transfer syntax for personal identity information such as private keys and certificates, etc.)
PKCS #15	Cryptographic Token Information Format Standard (ensures interoperability of multiple vendor implementations)
CAPI	Cryptographic Application Programming Interface standards
PC/SC Workgroup Specifications 1.0	Interoperability Specs for Smart Cards and PCs (platform and OS independent)
PC/SC Workgroup Specifications 2.0	Updated enhancements, including contactless (wireless RF) cards
ISO/IEC 7810	Identification Cards – physical characteristics
ISO/IEC 7811	ID Cards – Recording techniques
ISO/IEC 7812	ID Cards – Identification of issuers
ISO/IEC 7813	Financial transaction cards
ISO/IEC 7816	ID Cards with contacts
ISO/IEC 10373	ID Cards – Test Methods
ISO/IEC 10536	Contactless ID Cards – Close Coupled
ISO/IEC 14443	Contactless ID Cards – Proximity (Mifare cards) - 1-inch range
ISO/IEC 15693	Contactless ID Cards – Vicinity (I.CODE cards) - 5-inch range
Common Biometric Exchange Formats Framework (CBEFF)	CBEFF originally stood for Common Biometric Exchange File Format and was originally developed by the Biometric Consortium (BC). It was published by NIST as NISTR 6529. CBEFF defines a standard method for identifying and carrying biometric data. It describes a framework for defining data formats that facilitate the communication of biometric data. CBEFF does not specify the actual encoding of data (e.g., bits on a wire) but provides rules and requirements and the structure for defining those explicit data format specifications.
BioAPI	The BioAPI standard defines an Application Program Interface (API) and a Service Provider Interface (SPI) for standardizing the interaction between biometric-enabled applications and biometric sensor devices. The API provides a common method for applications to access biometric authentication technology without requiring application

This Table is (U//FOUO)	
Name	**Description**
	developers to have biometric expertise. The SPI allows the production of multiple BSPs (Biometric Service Providers) that may be used by an application without modification of that application, regardless of biometric technology. The BioAPI Consortium originally developed the BioAPI specification. The BioAPI Consortium is a group of over 50 organizations focused solely on furthering a standard biometric API. M1 has taken the resulting specification from the consortium and standardized it nationally as ANSI INCITS 358-2002. M1 has also contributed ANSI INCITS 358-2002 to SC 37 where it is currently a draft international standard.
Data Interchange Formats	A data interchange format specifies the low-level format for storing, recording, and transmitting biometric information. This biometric information may be unique to each biometric characteristic (e.g., fingerprint, iris, signature) and/or to each method of capture (e.g., photograph, capacitive sensor). In some technologies, this biometric information is called a template. M1.3 is currently working on projects dedicated to standards for the following formats.
Biometric Profiles	A biometric profile identifies a set of base biometric standards that apply to a single application or scenario. The profile then identifies the appropriate configurations, parameters, and choices for options provided within those specifications. The goal is to provide interoperability and consistent functionality and security across a defined environment. M1.4 is engaged in the following projects: Interoperability and Data Interchange—Biometric Based Verification and Identification of Transportation Workers Interoperability, Data Interchange and Data Integrity—Biometric Based Personal Identification for Border Management Point-of-Sale Biometric Verification/Identification SC 37 has defined a functional architecture that serves as part one of a multi-part standard. SC 37 is also working on the first profile of the standard titled Biometric Profile for Employees.
Biometric Evaluation Methodology	The Biometric Evaluation Methodology (BEM), Version 1.0, was designed to aid security evaluators who were attempting to evaluate biometric products against the Common Criteria (CC). The Common Evaluation Methodology (CEM) used in CC evaluations does not address the environmental, user population, and other issues that have an impact on a biometric implementation. The BEM specifically addresses these issues as they apply to biometric technology evaluations under the CC. Evaluators, certifiers and developers from Canada, U.K., GERMANY, U.S., Italy, Sweden, and others developed the BEM. Version 1.0 of BEM was released in August of 2002.
Biometrics Protection Profile	The CC is an effort of the US, Canada, and European countries to establish a common set of security criteria by which to evaluate IT products. This effort has resulted in an international standard (ISO/IEC 15408-1) for evaluating IT security products. The document that establishes the implementation-independent security requirements for a given category of product is called a Protection Profile. Currently, the DoD Biometrics Management Office (BMO) and the National Security Agency (NSA) are developing four Protection Profiles for biometrics products: Robustness Biometric PP for Verification Mode Basic Robustness Biometric PP for Verification Mode Medium Robustness Biometric PP for Identification Mode Basic Robustness Biometric PP for Identification Mode

This Table is (U//FOUO)	
Name	**Description**
Biometric API for JavaCard	The JavaCard Forum was established in 1997 to promote Java as the preferred programming language for multiple-application smart cards. A subset of the Java programming language was proposed for these cards and resulted in a standard for a JavaCard API. The JavaCard Forum has extended the JavaCard API to enroll and manage biometric data securely and facilitate a match on card capability with the Biometric API for JavaCard. The Biometric API manages templates, which are stored only in the card. During a match process, no sensitive information is sent off the card.
Common Data Security Architecture (CDSA), Human Recognition Services Module	The Human Recognition Services Module (HRS) is an extension of the Open Group's Common Data Security Architecture (CDSA). CDSA is a set of layered security services and a cryptographic framework that provides the infrastructure for creating cross-platform, interoperable, security-enabled applications for client-server environments. The biometric component of the CDSA's HRS is used in conjunction with other security modules (i.e., cryptographic, digital certificates, and data libraries) and is compatible with the BioAPI specification and CBEFF.
RFC 2413	Dublin Core Metadata For Resource Discovery
RFC 821	Simple Mail Transfer Protocol
RFC 822	Standard for the Format of ARPA Internet Text Messages
RFC 1421	Privacy Enhancement for Internet Electronic Mail: Part I: Message Encryption and Authentication Procedures
RFC 1422	Privacy Enhancement for Internet Electronic Mail: Part II: Certificate-Based Key Management
RFC 1423	Privacy Enhancement for Internet Electronic Mail: Part III: Algorithms, Modes, and Identifiers
RFC 1424	Privacy Enhancement for Internet Electronic Mail: Part IV: Key Certification and Related Services
RFC 1848	MIME Object Security Services
RFC 3852	Cryptographic Message Syntax (CMS)
RFC 3851	S/MIME v3.1 Message Specification
RFC 3850	S/MIME v3.1 Certificate Handling
RFC 2634	Enhanced Security Services for S/MIME
RFC 3854	Securing X.400 Content with S/MIME
RFC 3855	Transporting S/MIME Objects in X.400
RFC 3370	CMS Algorithms
RFC 2797	Certificate Management Messages over CMS
RFC 2616	Hypertext Transfer Protocol -- HTTP/1.1
RFC 2617	HTTP Authentication: Basic and Digest Access Authentication
RFC 2660	The Secure HyperText Transfer Protocol
RFC 2518	HTTP Extensions for Distributed Authoring -- WEBDAV
RFC 3744	WebDAV Access Control Protocol
RFC 2222	Simple Authentication and Security Layer (SASL)
RFC 2444	The One-Time-Password SASL Mechanism
RFC 2554	SMTP Service Extension for Authentication
RFC 1939	Post Office Protocol - Version 3
RFC 2449	POP3 Extension Mechanism

This Table is (U//FOUO)	
Name	**Description**
RFC 1734	POP3 AUTHentication command
RFC 3206	The SYS and AUTH POP Response Codes
RFC 3501	Internet Message Access Protocol (IMAP) - Version 4rev1
RFC 2195	IMAP/POP AUTHorize Extension for Simple Challenge/Response
RFC 1731	IMAP4 Authentication Mechanisms
RFC 2086	IMAP4 ACL extension
RFC 2228	FTP Security Extensions
RFC 2244	Application Configuration Access Protocol
X.400	Information Technology – Message Handling Systems (MHS) – Message Handling System and Service Overview
X.402	Information Technology – Message Handling Systems (MHS) – Overall Architecture
X.411	Information Technology – Message Handling Systems (MHS) – Message transfer system: Abstract Service Definition and Procedures
SDN.701	Message Security Protocol
ACP 120	Common Security Protocol (CSP)
PKCS #7	Cryptographic Message Syntax Standard
RFC 2246	The TLS Protocol v1.0
RFC 2817	Upgrading to TLS Within HTTP/1.1
RFC 2818	HTTP Over TLS
RFC 3546	TLS Extensions
RFC 3268	AES Ciphersuites for TLS
RFC 2829	Authentication Methods for LDAP
RFC 2830	LDAPv3 Extension for TLS
RFC 3377	LDAP v3 Technical Specification
RFC 2595	Using TLS with IMAP, POP3 and ACAP
RFC 3207	SMTP Service Extension for Secure SMTP over TLS
ISO/IEC 11586-1	Information technology -- Open Systems Interconnection -- Generic upper layers security: Overview, models and notation
ISO/IEC 11586-2	Information technology -- Open Systems Interconnection -- Generic upper layers security: Security Exchange Service Element (SESE) service definition
ISO/IEC 11586-3	Information technology -- Open Systems Interconnection -- Generic upper layers security: Security Exchange Service Element (SESE) protocol specification
ISO/IEC 11586-4	Information technology -- Open Systems Interconnection -- Generic upper layers security: Protecting transfer syntax specification
ISO/IEC 11586-5	Information technology -- Open Systems Interconnection -- Generic upper layers security: Security Exchange Service Element (SESE) Protocol Implementation Conformance Statement (PICS) proforma
ISO/IEC 11586-6	Information technology -- Open Systems Interconnection -- Generic upper layers security: Protecting transfer syntax Protocol Implementation Conformance Statement (PICS) proforma
ISO/IEC 7498-2	Data Communication Networks – Open Systems Interconnection (OSI) – Security, Structure and Applications – Security Architecture for Open Systems Interconnection for CCITT Applications

This Table is (U//FOUO)	
Name	**Description**
ISO/IEC 10745	Information Technology – Open Systems Interconnection – Upper Layers Security Model
CCITT X.800	Data Communication Networks – Open Systems Interconnection (OSI) – Security, Structure and Applications – Security Architecture for Open Systems Interconnection for CCITT Applications
ITU-T X.803	Information Technology – Open Systems Interconnection – Upper Layers Security Model
ITU-T X.830	Information technology -- Open Systems Interconnection -- Generic upper layers security: Overview, models and notation
ITU-T X.831	Information technology -- Open Systems Interconnection -- Generic upper layers security: Security Exchange Service Element (SESE) service definition
ITU-T X.832	Information technology -- Open Systems Interconnection -- Generic upper layers security: Security Exchange Service Element (SESE) protocol specification
ITU-T X.833	Information technology -- Open Systems Interconnection -- Generic upper layers security: Protecting transfer syntax specification
ITU-T X.834	Information technology -- Open Systems Interconnection -- Generic upper layers security: Security Exchange Service Element (SESE) Protocol Implementation Conformance Statement (PICS) proforma
ITU-T X.835	Information technology -- Open Systems Interconnection -- Generic upper layers security: Protecting transfer syntax Protocol Implementation Conformance Statement (PICS) proforma
[XML]	XML
	XML Schema
[XML-DSIG]	XML-DSIG
[XML-ENC]	XML-ENC
	XKMS
[SOAP]	SOAP
	WSDL
[SAML]	SAML
[XACML]	XACML
	UDDI
	SPML
	XCBF
	XCBF Token Profile
[WSS]	Web Services Security (WSS)
	WSS UsernameToken Profile
	WSS X.509 Certificate Token Profile
	Web Services Reliable Messaging
	ebXML Registry
	ebSOA
	WSDM
	XrML (eXtensible Rights Management Language)
	Web Application Security
	Digital Signature Services
	Security Services

This Table is (U//FOUO)	
Name	**Description**
	Web Services Distributed Management
[WSI-SEC]	Basic Security Profile Security Scenarios
	Basic Profile
	ANSI X9.84 (XCBF)
[XCMS]	ANSI X9.96 (XCMS)
	ANSI X9.73 (CMS)
	ITU-T X.509
	ISO 19092 (biometric formats)
[ID-FF]	ID-FF
[ID-SIS]	ID-SIS
[ID-WSF]	ID-WSF
	draft-lib-arch-soap-authn
[XML]	XML
	XML Schema
[XML-DSIG]	XML-DSIG
[XML-ENC]	XML-ENC
	XKMS
[SOAP]	SOAP
	WSDL
[SAML]	SAML
[XACML]	XACML
	UDDI
	SPML
	XCBF
	XCBF Token Profile
[WSS]	Web Services Security (WSS)
	WSS UsernameToken Profile
	WSS X.509 Certificate Token Profile
	Web Services Reliable Messaging
	ebXML Registry
	ebSOA
	WSDM
	XrML (eXtensible Rights Management Language)
	Web Application Security
	Digital Signature Services
	Security Services
	Web Services Distributed Management
[WSI-SEC]	Basic Security Profile Security Scenarios
	Basic Profile
	ANSI X9.84 (XCBF)

This Table is (U//FOUO)	
Name	**Description**
[XCMS]	ANSI X9.96 (XCMS)
	ANSI X9.73 (CMS)
	ITU-T X.509
	ISO 19092 (biometric formats)
[ID-FF]	ID-FF
[ID-SIS]	ID-SIS
[ID-WSF]	ID-WSF
	draft-lib-arch-soap-authn
FNBDT-210 (Signaling Plan)	This unclassified specification defines the signaling requirements for FNBDT operational modes. A secure overlay capable of interoperation with FNBDT compatible equipment on various similar or disparate networks is defined. Since the various networks will often have different lower-layer communications protocols, the FNBDT secure overlay specification specifies the higher-layer end-to-end protocols only. Appendices to this specification define operation using specific networks.
FNBDT-230 (Cryptography Specification)	This classified specification outlines details of the cryptography defined for FNBDT. Issues such as key generation, traffic encryption, and compromise recovery are specified in sufficient detail to allow interoperable implementation.
Proprietary extensions	The FNBDT signaling and cryptography specifications define interoperable branch points allowing vendors to implement proprietary modes. This allows vendors to take advantage of the basic FNBDT structure to add modes fulfilling specific needs. Legacy FNBDT implementations have used these branch points to implement custom cryptographic modes. Details of such modes are contained in vendor proprietary specifications.
Other specifications	Other interoperable FNBDT specifications have been suggested and are currently under consideration by the FNBDT Working Group. These additional documents would provide interoperable ways of implementing additional features such as non-Type 1 operation and key management.
FNBDT-210	Signaling Plan Revision 2.0
ITU V.150	Procedures for the end-to-end connection of V-series DCEs over and IP network
RFC 3550	RTP: A Transport Protocol for Real-Time Applications
RFC 3711	The Secure Real-time Transport Protocol (SRTP)
	Interoperability Specification For High Assurance Internet Protocol Encryptor (HAIPE) Devices
	Interoperability Specification For High Assurance Internet Protocol Encryptor (HAIPE) Devices
RFC-2401	Security Architecture for the Internet Protocol http://www.ietf.org/rfc/rfc2401.txt
	Security Architecture for the Internet Protocol http://www.ietf.org/internet-drafts/draft-ietf-ipsec-rfc2401bis-02.txt
RFC-2402	IP Authentication Header http://www.ietf.org/rfc/rfc2402.txt
	IP Authentication Header http://www.ietf.org/internet-drafts/draft-ietf-ipsec-rfc2402bis-07.txt
RFC-2406	IP Encapsulating Security Payload (ESP) http://www.ietf.org/rfc/rfc2406.txt

This Table is (U//FOUO)	
Name	**Description**
	IP Encapsulating Security Payload (ESP) http://www.ietf.org/internet-drafts/draft-ietf-ipsec-esp-v3-08.txt)
H.235	Security and encryption for H-series multimedia terminals
H.245	Call Control Protocol for multimedia communication: Series H
H.323	Packet-based multimedia communications: Series H
H.510	Mobility for H.323 multimedia systems and services
H.530	Symmetric security procedures for H.323 mobility in H.510
RFC 3262	SIP: Session Initiation Protocol
RFC 3310	Hypertext Transfer Protocol (HTTP) Digest Authentication Using Authentication and Key Agreement (AKA)
RFC 3313	Private Session Initiation Protocol (SIP) Extensions for Media Authorization
RFC 3323	A Privacy Mechanism for the Session Initiation Protocol (SIP)
RFC 3325	Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks
RFC 3329	Security Mechanism Agreement for the Session Initiation Protocol (SIP)
RFC 3435	Media Gateway Control Protocol
RFC 3525	Gateway Control Protocol
RFC 3761	The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM)
RFC 3762	Telephone Number Mapping (ENUM) Service Registration for H.323
RFC 3853	S/MIME Advanced Encryption Standard (AES) Requirement for the Session Initiation Protocol (SIP)
ETSI ES 201 733	European Technical Standards Institute, "Electronic Signature Formats", 2000. Available at http://webapp.etsi.org/exchangefolder/es_201733v010103p.pdf
ISO 13888-1	International Standards Organization, "IT security techniques -- Non-repudiation -- Part 1: General", 2004
ISO 13888-2	International Standards Organization, "Information technology -- Security techniques -- Non-repudiation -- Part 2: Mechanisms using symmetric techniques", 1998
ISO 13888-3	International Standards Organization, "Information technology -- Security techniques -- Non-repudiation -- Part 3: Mechanisms using asymmetric techniques", 1997.
SDN.801	SDN.801 addresses concepts, tools and mechanisms for implementation of access control (AC). SDN.801 should be used to gain both a global understanding of MISSI access control, and as a guide for implementing access control features in MISSI-compliant components. SDN.801 is designed to advance from general concepts that introduce access control to more detailed information on access control tools, mechanisms, and processes as they apply to real-world communication systems.
ANSI INCITS 359-2004	This standard describes Role Based Access Control (RBAC) features that have achieved acceptance in the commercial marketplace. It includes a reference model and functional specifications for the RBAC features defined in the reference model. RBAC has become the predominant model for advanced access control because it reduces the complexity and cost of security administration in large networked applications. Many information technology vendors have incorporated RBAC into their product line, and the technology is finding applications in areas ranging from health care to defense, in addition to the mainstream commerce systems for which it was designed. The National Institute of Standards and Technology (NIST) initiated the development of the standard via the

This Table is (U//FOUO)	
Name	**Description**
	INCITS fast track process.
XACML 1.0	XACML is an XML-based language, or schema, designed specifically for creating policies and automating their use to control access to disparate devices and applications on a network.
RFC 3157	This document identifies a set of requirements for credential mobility. Using SACRED protocols, users will be able to securely move their credentials between different locations, different Internet devices, and different storage media as needed.
Extensible Access Control markup Language (XACML)	XACML provides fine-grained control of authorized activities, the effect of characteristics of the access requestor, the protocol over which the request is made, authorization based on classes of activities, and content introspection.
Routing Policy Specification Language (RPSL)	RPSL allows a network operator to be able to specify routing policies at various levels in the Internet hierarchy. Policies can be specified with sufficient detail in RPSL so that low-level router configurations can be generated from them. RPSL is extensible; new routing protocols and new protocol features can be introduced at any time.
Rei	A declarative policy language for describing policies over actions. It is possible to write Rei policies over ontologies in other semantic web languages.
KeyNote	KeyNote provides a simple language for describing and implementing security policies, trust relationships, and digitally signed credentials.
SDN.801	SDN.801 provides guidance for implementing access control concepts using both public key certificates and attribute certificates.
Security Assertion Markup Language (SAML)	SAML is an XML framework for exchanging authentication and authorization information.
Ponder	Ponder is a language for specifying management and security policies for distributed systems.
KAoS	KAoS policy services allow for the specification, management, conflict resolution, and enforcement of policies within domains.
LDAP	LDAP is an Internet protocol used to look up information from a LDAP server or directory. LDAP servers index all the data in their entries, and "filters" may be used to select just the information you want. "Permissions" and “authentications” can be set by the administrator to allow only certain people to access the LDAP database, and optionally keep certain data private. Reference http://www.ldap-directory.org/rfc-ldap for a list of LDAP RFCs.
File Transfer Protocol (FTP)	File Transfer Protocol (FTP), a standard Internet protocol, is the simplest way to exchange files between computers on the Internet. FTP is an application protocol that uses the Internet's TCP/IP protocols. Reference RFC959: http://www.w3.org/Protocols/rfc959/
Common Open Policy Service (COPS)	The Common Open Policy Service (COPS) protocol is a simple query and response protocol that can be used to exchange policy information between a policy server (PDP) and its clients (PEPs). Reference http://www.networksorcery.com/enp/protocol/cops.htm for a list of COPS related RFCs
Microsoft’s SMS	SMS provides a solution for change and configuration management for the Microsoft platform, enabling organizations to provide relevant software and updates to users quickly and cost effectively.
Telnet	The Telnet program allows you to connect your PC to a server on the network using a

This Table is (U//FOUO)	
Name	**Description**
	username and password. You can then enter commands through the Telnet program, and they will be executed as if you were entering them directly on the server console.
SSL	SSL is designed to make use of TCP as a communication layer to provide a reliable end-to-end secure and authenticated connection between two points over a network.
TLS	RFC2246: The primary goal of the Transport Layer Security (TLS) Protocol is to provide privacy and data integrity between two communicating applications. The protocol is composed of two layers: the TLS Record Protocol and the TLS Handshake Protocol. At the lowest level, layered on top of some reliable transport protocol (e.g., TCP), is the TLS Record Protocol. The TLS Record Protocol provides connection security that provides confidentiality and integrity. TLS is designed as a successor to SSL and is sometimes called SSL V3.0.
IPsec	RFC 2401: Internet Protocol Security (generally shortened to IPsec) is a framework of open standards that provides data confidentiality, data integrity, and data authentication between participating peers at the IP layer. IPsec can be used to protect one or more data flows between IPsec peers.
X.500	X.500 is a CCITT protocol that is designed to build a distributed, global directory. It offers decentralized maintenance, searching capabilities, single global namespace, structured information framework, and a standards-based directory.
Finger, whois, domain name	These are very simple directory formats that are also in use.
RFC 2386	A Framework for QoS-Based Routing in the Internet
RFC 2676	QoS Routing Mechanisms and OSPF Extensions
SAML Core	E. Maler et al. Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML). OASIS, September 2003. Document ID oasis-sstc-saml-core-1.1. http://www.oasis-open.org/committees/security/.
SAML Gloss	E. Maler et al. Glossary for the OASIS Security Assertion Markup Language (SAML). OASIS, September 2003. Document ID oasis-sstc-saml-glossary-1.1.http://www.oasis-open.org/committees/security/.
SAMLSec	E. Maler et al. Security Considerations for the OASIS Security Assertion Markup Language (SAML), OASIS, September 2003, Document ID oasis-sstc-saml-sec-consider-1.1. http://www.oasis-open.org/committees/security/
SAMLReqs	Darren Platt et al., SAML Requirements and Use Cases, OASIS, April 2002, http://www.oasis-open.org/committees/security/.
SAMLBind	E. Maler et al. Bindings and Profiles for the OASIS Security Assertion Markup Language (SAML). OASIS, September 2003. Document ID oasis-sstc-saml-bindings-1.1. http://www.oasis-open.org/committees/security/.
SPML – Service Provisioning Markup Language	SPML is intended to facilitate the creation, modification, activation, suspension, and deletion of data on managed Provision Service Targets (PSTs). It is the only real standard of import that deals explicitly with the act of provisioning. Provisioning is a core component of Identity Management, but unfortunately most of the standards work has been in the direction of privilege management. http://www.oasis-open.org/committees/documents.php
SPML-Bind	OASIS Provisioning Services Technical Committee., SPML V1.0 Protocol Bindings, http://www.oasis-open.org/apps/org/workgroup/provision/download.php/1816/draft-pstc-bindings-03.doc, OASIS PS-
XACML – eXtensible Access Control Markup	From http://www.oasis-open.org/committees/download.php/2713/Brief_Introduction_to_XACML.html

This Table is (U//FOUO)	
Name	**Description**
Language	
ID-FF	Identity Federation Framework Available from http://www.projectliberty.org/
ID WSF	Identity Web Service Framework Available from http://www.projectliberty.org/
ID SIS	Identity Services Interface Specifications Available from http://www.projectliberty.org/
RFC3281	S. Farrell, R. Housley, "An Internet Attribute Certificate Profile for Authorization", IETF RFC, April 2002
ISO/IEC 9594-8	ITU-T Rec. X.509 (2000) | ISO/IEC 9594-8 The Directory: Authentication Framework
S/MIME	Ramsdell, B., "S/MIME Version 3 Message Specification", RFC2633, June 1999
MIME	Freed, N., Borenstein, N., "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996.
CMS	Housley, R., "Cryptographic Message Syntax", RFC 3369, June 1999.
X9.69	Framework for Key Management Extensions. This standard defines specific key management methods for controlling and handling keys.
X9.73	Cryptographic Message Syntax (CMS) The Constructive Key Management technique (CKM), described in ANS X9.69, is used to encrypt objects. It may be used with CMS to encrypt a message (as the object) to a set of users sharing a common set of values (known as key components).
X9.42	Key Agreement of Symmetric Keys using Discrete Logarithm Cryptography.
X9.44	Key Establishment Using Factoring-Based Public Key Cryptography.
FIPS PUB 140-2 ANNEX D	Security Requirements for Cryptographic Modules Annex D: Approved Key Establishment Techniques Annex D provides a list of the FIPS Approved key establishment techniques applicable to FIPS PUB 140-2.
XKMS	XML Key Management Specification (XKMS) http://csrc.nist.gov/cryptval/140-2.htm
FIPS 171	Symmetric Key Establishment Techniques National Institute of Standards and Technology, Key Management using ANSI X9.17, Federal Information Processing Standards Publication 171, April 27, 1992. http://csrc.nist.gov/publications/fips/fips171/fips171.txt
EKMS 208	EKMS Key Distribution Functional Standard. National Security Agency, Director, National Security Agency, Ft. George G. Meade, MD. 20755-6734.
EKMS 215	EKMS Communications Requirements Standard. National Security Agency, Director, National Security Agency, Ft. George G. Meade, MD. 20755-6734.
EKMS 301	EKMS Types Dictionary Standard. National Security Agency, Director, National Security Agency, Ft. George G. Meade, MD. 20755-6734.
EKMS 302	EKMS Key Distribution Data Standard. National Security Agency, Director, National Security Agency, Ft. George G. Meade, MD. 20755-6734.
EKMS 311	EKMS ACCORDION 1.3 Length Indicator and Binding Code Specification. National Security Agency, Director, National Security Agency, Ft. George G. Meade MD. 20755-

This Table is (U//FOUO)	
Name	**Description**
	6734.
EKMS 603	Interface Specification for the Data Transfer Device AN/CYZ-10. National Security Agency, Director, National Security Agency, Ft. George G. Meade, MD. 20755-6734.
	[XAdES] J.C. Cruellas, G. Karlinger, K. Sankar XML Advanced Electronic Signatures; W3C Note 20 February, 2003 http://www.w3.org/TR/XAdES/
XML	Bray, T., Paoli, J., Sperberg-McQueen, C. M., Maler, E., “Extensible Markup Language (XML) 1.0 (Second Edition),” W3C Recommendation 6 October, 2000.
XMLENC	Eastlake, D., Reagle, J., Imamura, T., Dillaway, B., Simon, E., “XML Encryption Syntax and Processing,” W3C Recommendation 10 December, 2002.
XMLSIG	Eastlake, D., Reagle, J., Solo D., “(Extensible Markup Language) XML-Signature Syntax and Processing,” RFC 3075, March, 2002.
XMLSEC	Mactaggart, M., “Enabling XML Security: An introduction to XML encryption and XML signature,” http://www-106.ibm.com/developerworks/xml/library/s-xmlsec.html/index.html.
	KMI-2200, dated July, 2004
DES	U.S. Data Encryption Standard (DES) in accordance with U.S. FIPS PUB 46-2 and ANSI X3.92
AES	U.S. Advanced Encryption Standard (AES) in accordance with U.S. FIPS PUB 197 (256-bit keys supported)
CAST block cipher	CAST block cipher in accordance with RFC 2144 (64-bit, 80-bit, and 128-bit variations are supported)
Triple-DES	Triple-DES in accordance with ANSI X9.52 (3-key variant for an effective key size of 168-bits is supported)
RC2®	RC2® in accordance with RFC 2268 (40-bit and 128-bit variations are supported)
IDEA	IDEA as listed in the ISO/IEC 9979 Register of Cryptographic Algorithms (128-bit supported)
RSA	RSA in accordance with Public Key Cryptographic Standards (PKCS) specification PKCS#1 Version 2.0, ANSI X9.31, IEEE 1363, ISO/IEC 14888-3 and U.S. FIPS PUB 186-2 (1024-bit, 2048-bit, 4096-bit and 6144-bit supported)
DSA	DSA in accordance with the Digital Signature Standard, U.S. FIPS PUB 186-2, ANSI X9.30 Part 1, IEEE P1363 and ISO/IEC 14888-3 (1024-bit supported)
ECDSA	ECDSA in accordance with ANSI X9.62, IEEE P1363, ISO/IEC 14888-3 and U.S. FIPS PUB 186-2 (192-bit default)
SHA-1, SHA-256, SHA-384, and SHA-512	SHA-1, SHA-256, SHA-384 and SHA-512 in accordance to U.S. FIPS PUB 180-2 and ANSI X9.30 Part 2
MD5 Message-Digest algorithm	MD5 Message-Digest algorithm in accordance with RFC 1321
MD2 Message-Digest algorithm	MD2 Message-Digest algorithm in accordance with RFC 1319
RIPEMD-160	RIPEMD-160 in accordance with ISO/IEC 10118-3:1998
RSA key transfer	RSA key transfer in accordance with RFC 1421 and RFC 1423 (PEM), PKCS#1 Version 2.0, IEEE P1363
Diffie-Hellman key agreement	Diffie-Hellman key agreement in accordance with PKCS#3
Simple Public-Key	Simple Public-Key GSS-API Mechanism (SPKM) authentication and key agreement in

This Table is (U//FOUO)	
Name	**Description**
GSS-API Mechanism (SPKM) authentication and key	accordance with RFC 2025, ISO/IEC 9798-3 and U.S. FIPS PUB 196
SSL v3 and TLS v1	SSL v3 and TLS v1 in accordance with RFC 2246
MAC	MAC in accordance with U.S. FIPS PUB 113 (for DES-MAC) and X9.19
HMAC	HMAC in accordance with RFC 2104
Pseudo random number generator	Pseudo random number generator in accordance with ANSI X9.17 (Appendix C) and FIPS 186-2
Version 3 public-key certificates and Version 2 CRLs	Version 3 public-key certificates and Version 2 CRLs in accordance with ITU-T X.509 Recommendation and ISO/IEC 9594-8 (4th edition, 2000 as well as earlier editions)
Version 3 public-key certificate and Version 2 CRL extensions	Version 3 public-key certificate and Version 2 CRL extensions in accordance with RFC 2459 and RFC 3280
Version 3 public-key certificate and Version 2 CRL extensions	Version 3 public-key certificate and Version 2 CRL extensions in accordance with U.S. FPKI X.509 Certificate and CRL Extensions Profile
Version 3 public-key certificate and Version 2 CRL extensions	Version 3 public-key certificate and Version 2 CRL extensions in accordance with NIST X.509 Certificate and CRL Extensions Profile for the Common Policy
Version 3 "Qualified" certificates	Version 3 "Qualified" certificates in accordance with RFC 3039 and ETSI TS 101 862
Version 3 public-key certificates and Version 2 CRLs	Version 3 public-key certificates and Version 2 CRLs in accordance with de-facto standards for Web browsers and servers
WTLS Certificate support in accordance with WAP WTLS Version 1.1.	WTLS Certificate support in accordance with WAP WTLS Version 1.1. (certificate issuance)
RSA algorithm identifiers and public key formats	RSA algorithm identifiers and public key formats in accordance with RFC 1422 and 1423 (PEM) and PKCS#1
Online Certificate Status Protocol, version 2. Working document of the IETF	Online Certificate Status Protocol, version 2. Working document of the IETF RFC 2560.
Standard file envelope format	Standard file envelope format based on Internet RFC 1421 (PEM)
PKCS#7 Version 1.5 based on RFC	PKCS#7 Version 1.5 based on RFC 2315 and Cryptographic Message Syntax (CMS) based on RFC 3369 and 3370

This Table is (U//FOUO)	
Name	**Description**
2315 and Cryptographic Message Syntax (CMS)	
S/MIME Version 2	S/MIME Version 2 based on RFC 2311
On-line GSS-API public key implementation mechanism using SPKM	On-line GSS-API public key implementation mechanism using SPKM in accordance with Internet RFC 2025 and SPKM entity authentication in accordance with FIPS 196
SSL v3 and TLS v1	SSL v3 and TLS v1 in accordance with RFC 2246
LDAP Version 2	LDAP Version 2 in accordance with RFC 1777 and RFC 2559
LDAP Version 3	LDAP Version 3 in accordance with RFC 2251-2256
Private key storage	Private key storage in accordance with PKCS#5 and PKCS#8
Secure Exchange Protocol (SEP)	Secure Exchange Protocol (SEP), built using Generic Upper Layers Security (GULS) standards ITU-T Recs. X.830, X.831, X.832 and ISO/IEC 11586-1, 11586-2, 11586-3 (SEP continues to be supported for backward compatibility only)
PKIX-CMP	PKIX-CMP in accordance with RFC 2510 and PKIX-CRMF in accordance with RFC 2511
PKCS 7/10	PKCS 7/10 (for Web based clients and VPN solutions)
Cisco Certificate Enrollment Protocol (CEP)	Cisco Certificate Enrollment Protocol (CEP) (for VPN solutions)
Hardware cryptographic interface	Hardware cryptographic interface in accordance with PKCS#11
Generic Security Services API (GSS-API)	Generic Security Services API (GSS-API) in accordance with RFC 1508 and 1509
IDUP-GSS-API	IDUP-GSS-API in accordance with Internet Draft draft-ietf-cat-idup-gss-08.txt
SNMPv3	The Simple Network Management Protocol, version 3 is the latest version of the IETF standard for managing network devices. Version 3 includes authentication and authorization, so is considered much more secure than previous versions. SNMP is widely implemented, but has some significant restrictions because of its very simple structure.
TFTP	The Trivial File Transfer Protocol (TFTP), as defined by IETF RFC 1350, is a very simple file transfer protocol that can be implemented in very small systems, such as firmware. It implements no authentication whatsoever and consequently is usable only in the most benign, protected environments.
DHCP	The Dynamic Host Control Protocol (DHCP) is defined by IETF RFC 2131 and modified by a host of other RFCs. It allows a machine, which at network initialization time does not know its own IP address, to request allocation of an IP address from a server and receive network configuration data sufficient to communicate on an IP network.
SM Spec	Signed Manifest Specification, The Open Group SM Spec Signed Manifest Specification, The Open Group, 1997. http://www.opengroup.org/pubs/catalog/c707.htm
CIM	The Distributed Management Task Force (DMTF) originally developed the Common Information Model (CIM) to provide a data model for integrating management across SNMP, the Desktop Management Interface (DMI) (another part of WBEM), Common

This Table is (U//FOUO)	
Name	**Description**
	Management Information Protocol (CMIP or ISO 9596) (for telecom devices) and private applications. CIM is part of the DMTF's overall Web-based Enterprise Management (WBEM) initiative. WBEM includes CIM as the data definition, XML as the transport/encoding method, and HTTP as the access mechanism. CIM is an object-oriented data model for describing managed elements across the enterprise, including systems, networks, and applications. The CIM schema provides definitions for servers, desktops, peripherals, operating systems, applications, network components, users, and others along with details of each. One of the main functions CIM offers is the ability to define the associations between components. CIM's object-oriented approach makes it easier to track the relationships and interdependencies between managed objects. WBEM/CIM proponents promote this as a key advantage over SNMP.
WBEM	The Web-Based Enterprise Management (WBEM) standard is an initiative by the DMTF to develop a broader enterprise management structure than SNMP. The DMTF is an industry coalition that is developing an enterprise management framework for computer systems that is richer than SNMP, the WBEM standards.
SMBIOS	The System Management Basic I/O System (SMBIOS) is a DMTF standard for making firmware-level information available via a CIM model on computer systems.
Intel PXE Specification	The Intel-developed Preboot eXecution Environment (PXE) specification defines an OS-independent firmware-level mechanism for booting from a variety of media, including the network, using standard protocols. ftp://download.intel.com//labs/manage/wfm/download/pxespec.pd
Intel PXE BIS Specification	The Intel PXE Boot Integrity Services is an extension to the Intel PXE specification that provides for PKI-based authentication of the server to the booting client. ftp://download.intel.com//labs/manage/wfm/download/bisspec.zip
EPC Tag Data Specification Version 1.1	Identifies the specific encoding schemes for a serialized version of the EAN.UCC Global Trade Item Number (GTIN®), the EAN.UCC Serial Shipping Container Code (SSCC®), the EAN.UCC Global Location Number (GLN®), the EAN.UCC Global Returnable Asset Identifier (GRAI®), the EAN.UCC Global Individual Asset Identifier (GIAI®), and a General Identifier (GID)
900 MHz Class 0 Radio Frequency (RF) Identification Tag Specification.	This document specifies the communications interface and protocol for 900 MHz Class 0 operation. It includes the RF and tag requirements and provides operational algorithms to enable communications in this band.
13.56 MHz ISM Band Class 1 Radio Frequency (RF) Identification Tag Interface Specification.	This specification defines the communications interface and protocol for 13.56 MHz Class 1 operation. It also includes the RF and tag requirements to enable communications in this band.
860MHz – 930 MHz Class 1 Radio Frequency (RF) Identification Tag Radio Frequency & Logical Communication Interface Specification	This document specifies the communications interface and protocol for 860 – 930 MHz Class 1 operation. It includes the RF and tag requirements to enable communications in this band.
Physical Markup	The PML Core specification establishes a common vocabulary set to be used within the

This Table is (U//FOUO)	
Name	**Description**
Language (PML)	EPCglobal Network. It provides a standardized format for data captured by readers. This specification also includes XML Schema and Instance files for your reference.
ISO/IEC 15963:2004	Information technology – Radio frequency identification for item management – Unique identification for RF tags
ISO/IEC 18000-4:2004	Information technology – Radio frequency identification for item management – Part 4: Parameters for air interface communications at 2.45 GHz
ISO/IEC 18000-6:2004	Information technology – Radio frequency identification for item management – Part 6: Parameters for air interface communications at 860 MHz to 960 MHz
ISO/IEC 18000-7:2004	Information technology – Radio frequency identification for item management – Part 7: Parameters for active air interface communications at 433 MHz
FIPS 140-2	Security Requirements for Cryptographic Modules
SNMPv3	The Simple Network Management Protocol, version 3 is the latest version of the IETF standard for managing network devices. Version 3 includes authentication and authorization, so it is considered much more secure than previous versions. SNMP is widely implemented, but has some significant restrictions because of its very simple structure.
ISO/IEC 15408-1:1999	Information technology – Security techniques – Evaluation criteria for IT security – Part 1: Introduction and general model
ISO/IEC 15408-2:1999	Information technology – Security techniques – Evaluation criteria for IT security – Part 2: Security functional requirements
ISO/IEC 15408-3:1999	Information technology – Security techniques – Evaluation criteria for IT security – Part 3: Security assurance requirements
CLF	Common Log Format. Typically, the information is presented in plain ASCII without special delimiters to separate the different fields. See http://www.ietf.org
ELF	Extended Log Format
IDMEF	Intrusion Detection Message Exchange Format ietf.org/html.charters/idwg-charter.html The IETF's Intrusion Detection Working Group (IDWG) is developing message formats and procedures for sharing messages between intrusion detection systems and the SEM systems that manage them. The IDMEF requirements were posted in an Internet Draft in October, 2002, along with a draft of the Intrusion Detection Exchange Protocol (IDXP). In January, 2003, an Internet Draft was submitted for IDMEF that included an XML implementation. This initiative is still in development and it's future is not determined
RFC 1155,	Structure of Management Information
RFC 1156	Management Information Base (MIB-I)
RFC 1157	SNMP
RFC 1187	Bulk table retrieval
RFC 1212	Concise MIB definitions
RFC 1213	Management Information Base (MIB-II)
RFC 1215	Traps
RFC 1227	SNMP Multiplex (SMUX)
RFC 1228	SNMP-DPI
RFC 1229	Generic-interface MIB extensions
RFC 1239	Reassignment of MIBs

This Table is (U//FOUO)	
Name	**Description**
RFC 1243	AppleTalk MIB
RFC 1248	OSPF MIB
1230 IEEE 802.4	Token Bus MIB
1231 IEEE 802.5	Token Ring MIB
ISO 8824-1	Abstract Syntax Notation One (ASN.1): Specification of basic notation
ISO 8824-2	Abstract Syntax Notation One (ASN.1): Information object specification
ISO 8824-3	Abstract Syntax Notation One (ASN.1): Constraint specification
ISO 8824-4	Abstract Syntax Notation One (ASN.1): Parameterization of ASN.1 specifications
ISO 8825-1	ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)
ISO 8825-4	ASN.1 encoding rules: XML Encoding Rules (XER)
The Table is (U//FOUO)	

(U//FOUO) APPENDIX C: TV-2 FOR IA

(U) Current technologies do not provide sufficient capabilities to satisfactorily enable required GIG IA capabilities. Therefore, new technologies and standards will need to be developed. Table C-1 provides an initial view of the required technologies and when they are expected to mature. This table is a summary of the figures shown in Section 3 and the same cautions apply. Only the gaps and recommendations are discussed for technologies needed to meet the 2008 GIG IA objectives as discussed in the Transition Strategy (RCD Volume I). The discussion is further limited to technologies that are deemed risky, either because no work is currently ongoing, or because ongoing development effort will not be completed in time to deploy for 2008. In some cases, gaps and recommendations are summarized for technologies needed for 2012 and beyond, but only in cases where technology development efforts must begin now in order to meet those milestone dates.

Table C-1: (U//FOUO) TV-2 for IA

This Table is (U//FOUO)

Technology/Standard	**Short term 2006 (or earlier)**	**Mid term 2008**	**Long term 2010+**
Assured Information Sharing			
Authentication Session Score Standard	Standard defined	Begin Compliance with Authentication Standard	
Authentication Confidence	Standard defined	Begin Compliance with Authentication Standard	
Authentication Tokens Hardware Tokens (CAC)	Standard defined	IA Enhanced CAC	
Biometrics	Medium and High Assurance PP's		
Pilots using Policy-driven AC Mechanisms - IDs, Privs and I&A SoM, with Manual Override Support	Pilots begin	Traditional Access Control Process	
Metadata Standard	Standard defined	Labeling Standard Ratified	
Initial IA Metadata Creation Tools	Pilots begin	Labeled data pilots	
Cryptobinding of Metadata			Standard defined
CDS Browse/Query	Standard defined	Improved CDS filtering	
CDS Collaboration Suite		Secure chat, e-mail w/ attachments	
CDS Databases		Bi-directional discovery and retrieval	
Single Sign-on	Standard defined	NCES IOC Single Sign-on	
Special Purpose Trusted Platforms	Standard defined	MILS pilots	

This Table is (U//FOUO)			
Technology/Standard	**Short term 2006 (or earlier)**	**Mid term 2008**	**Long term 2010+**
Multi-Purpose Trusted Platforms		Standard defined	High assurance platforms
Simple Trusted Applications	Standard defined		
Protection Profiles for Medium and High Assurance Access Control Mechanisms		Initial Authentication Infrastructure Standard for trusted software development Object sanitization research	
Highly Available Enterprise			
Policy-based Network Management	Network control functions automated within a single domain		
IA Policy-based Routing	Exchange of routing across tunnels (red/red routing exchange)	IA policy based routing implemented Initial support for mobile/tactical IA policy-based routing	
HAIPEv2 Products		Edge-to-edge enterprise boundary protection eliminating red gateways	
TRANSEC (Research TBD)		TRANSEC for wireless/radio links	
GIG ID Management Standard	Standard defined	Strong I&A of network admins	
Authentication Tokens Hardware Tokens		Standard defined	
Integrity/Confidentiality of Network Management & Control	Standard defined	Confidentiality and integrity of management & control	
Operational-based Resource Allocation		Limited support for end-to-end resource allocation	Operational-based resource allocation implemented
Cyber Situational Awareness and Network Defense			
Host-based IDS			Standard defined
Network-based IDS	Standard defined	Standardized sensors	
Anomaly Detection	Standard defined		
UDOP	Standard defined	Initial UDOP tools available	
NETCOP		Standard defined	
Network Mapping	Standard defined		
Vulnerability Scanning	Standard defined		

This Table is (U//FOUO)

Technology/Standard	Short term 2006 (or earlier)	Mid term 2008	Long term 2010+
Host-based IPS		Initial automated analysis tools	
Network-based IPS	Standard defined	Research into automated response capability	
Traceback		Standard defined	
Correlation			Standard defined
Misuse Detection		Research misuse detection and intrusion detection in the Black Core	Standard defined
User Activity Profiling	Standard for collection, processing, & exchange of IA sensor data on IPv4 networks	Standard for collection, processing, & exchange of IA sensor data on IPv6 networks Devices capable of reporting their location available	
Assured Enterprise Management and Control			
User Identity Management	Standard accepted	All human users identified in accordance with GIG ID standard	Full implementation
Role-based Privileges	Privilege management standard ratified		
Privilege Management Infrastructure		Initial privilege management service	
OTNK	Benign fill	OTNK for wired and wireless products	ONTK for low bandwidth devices ONTK for coalition forces
Certificate Management		Identities in certificates comply with GIG ID standard	
Universal Configuration Management	Configuration management standards ratified		
Trusted Software Download	Secure software download Policy standards ratified	GIG-wide IA agents with trusted software download Initial policy infrastructure including deconfliction and synchronization	
Audit Format Standard	Audit format and exchange standard ratified		

This Table is (U//FOUO)			
Technology/Standard	**Short term 2006 (or earlier)**	**Mid term 2008**	**Long term 2010+**
Audit Aggregation & Analysis Standard		Compliance with audit standard, initial audit tools available	
This Table is (U//FOUO)			

15991