



## **Security for Management Interfaces to Network Elements 2.0**

**IA # OIF-SMI-03.1**

*October 8, 2012*

Implementation Agreement created and approved  
by the Optical Internetworking Forum  
[www.oiforum.com](http://www.oiforum.com)

---

The OIF is an international non-profit organization with over 90 member companies, including the world's leading carriers and vendors. Being an industry group uniting representatives of the data and optical worlds, OIF's purpose is to accelerate the deployment of interoperable, cost-effective and robust optical internetworks and their associated technologies. Optical internetworks are data networks composed of routers and data switches interconnected by optical networking elements.

With the goal of promoting worldwide compatibility of optical internetworking products, the OIF actively supports and extends the work of national and international standards bodies. Working relationships or formal liaisons have been established with IEEE 802.1, IEEE 802.3ba, IETF, IP-MPLS Forum, IPv6 Forum, ITU-T SG13, ITU-T SG15, MEF, ATIS-OPTXS, ATIS-TMOC, TMF and the XFP MSA Group.

For additional information contact:  
The Optical Internetworking Forum, 48377 Fremont Blvd.,  
Suite 117, Fremont, CA 94538  
+1 510 492 4040  
info@oiforum.com  
  
www.oiforum.com

# Security for Management Interfaces to Network Elements 2.0

---

**ABSTRACT:** This Implementation Agreement lists objectives for securing OAM&P interfaces to a network element and then lists guidelines for using security systems (e.g., IPsec or TLS) to protect these interfaces. It summarizes how well each of the systems, used as described, satisfies the objectives. It updates and obsoletes *Security for Management Interfaces to Network Elements* (OIF-SMI-01.0) and the *Addendum to the Security for Management Interfaces to Network Elements* (OIF-SMI-02.1).

---

## TECHNICAL EDITOR

Richard Graveman, RFG Security  
for Department of Defense  
15 Park Avenue  
Morristown, NJ 07960 USA  
+1 973 984 8780  
rfg@acm.org

## WORKING GROUP CHAIRS

Rémi Theillaud, Marben Products  
Evelyne Roch, Ciena Corporation  
Doug Zuckerman, Telcordia Technologies

**Notice:** This Technical Document has been created by the Optical Internetworking Forum (OIF). This document is offered to the OIF Membership solely as a basis for agreement and is not a binding proposal on the companies listed as resources above. The OIF reserves the rights to at any time to add, amend, or withdraw statements contained herein. Nothing in this document is in any way binding on the OIF or any of its members.

The user's attention is called to the possibility that implementation of the OIF implementation agreement contained herein may require the use of inventions covered by the patent rights held by third parties. By publication of this OIF implementation agreement, the OIF makes no representation or warranty whatsoever, whether expressed or implied, that implementation of the specification will not infringe any third party rights, nor does the OIF make any representation or warranty whatsoever, whether expressed or implied, with respect to any claim that has been or may be asserted by any third party, the validity of any patent rights related to any such claim, or the extent to which a license to use any such rights may or may not be available or the terms hereof.

© 2012 Optical Internetworking Forum

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction other than the following. (1) the above copyright notice and this paragraph must be included on all such copies and derivative works, and (2) this document itself may not be modified in any way, such as by removing the copyright notice or references to the OIF, except as needed for the purpose of developing OIF Implementation Agreements.

By downloading, copying, or using this document in any manner, the user consents to the terms and conditions of this notice. Unless the terms and conditions of this notice are breached by the user, the limited permissions granted above are perpetual and will not be revoked by the OIF or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE OIF DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY, TITLE OR FITNESS FOR A PARTICULAR PURPOSE.

## Table of Contents

1. Introduction.....	1
1.1 Problem Statement.....	1
1.2 Scope .....	1
1.3 Relationship to other Standards Development Organizations (SDOs).....	3
1.4 Acknowledgements .....	5
1.5 Outline of the Implementation Agreement.....	5
1.6 How to Use this Implementation Agreement .....	6
1.7 Document Organization.....	6
2. Terminology and Acronyms .....	7
2.1 Keywords.....	7
2.2 Terminology .....	7
2.3 Acronyms.....	8
3. Threats and Security Objectives .....	10
3.1 Confidentiality .....	11
3.2 Data Integrity.....	11
3.3 Key Management.....	12
3.4 Authentication .....	12
3.5 Negotiation and Policy Enforcement.....	13
3.6 Non-Repudiation .....	13
3.7 Access Control.....	13
3.8 Audit and Event Logging .....	14
3.9 Denial of Service .....	14
3.10 Traffic Analysis .....	15
4. Management Interfaces and Protocol Stacks .....	15
4.1 Protocol Stacks and Security .....	15
4.2 Protocol Stacks and VPNs.....	16
4.3 Management Interfaces and Security Protocols .....	17
5. Security Systems and Specifications .....	19
5.1 IPsec.....	19
5.2 Transport Layer Security (TLS) .....	21
5.3 SNMP .....	25
5.4 Secure Shell and Integrated Security for Management Systems .....	28
5.5 Secure Web-Based Management.....	31
5.6 Other Protocols Supporting Security .....	40
6. Objectives Satisfied by Security Systems.....	41
7. Summary .....	43
8. References.....	43
8.1 Normative References .....	43
8.2 Informative References.....	48
Appendix A: Glossary.....	51
Appendix B: OIF Members When the Document Was Approved .....	51

## *List of Figures*

Figure 1: Network Management Security Reference Model (from [T1M1]).	.....	<b>Error!</b>
<b>Bookmark not defined.</b>		
Figure 2: Typical Protocol Stacks for Management Interfaces.	.....	15
Figure 3: Protocol Stacks and Security.	.....	16
Figure 4: Protocol Stacks Including a Layer 3 VPN.	.....	17
Figure 5: AH and ESP in Transport Mode.	.....	20
Figure 6: AH and ESP in Tunnel Mode.	.....	20

## *List of Tables*

Table 1: Applicability of Security Solutions to Different Interfaces.	.....	18
Table 2: Web Services Security Standards from the IETF.	.....	35
Table 3: Web Services Security Standards from the W3C.	.....	36
Table 4: Web Services Security Standards from OASIS.	.....	38
Table 5: Web Services Security Standards from ANSI, NIST, and ITU-T.	.....	40
Table 6: Applicability of Security Solutions to Different Interfaces.	.....	41

# Security for Management Interfaces to Network Elements 2.0

## 1. Introduction

### 1.1 Problem Statement

The OIF has defined security mechanisms for its UNI and E-NNI Implementation Agreements, which describe how network elements (NEs) use various control protocols for signaling, routing, and discovery. NEs, however, typically have one or more (in some cases many) OAM&P interfaces used for network management, billing and accounting, configuration, error logging, maintenance, and other administrative activities. Remote access to a NE through these OAM&P interfaces is frequently a requirement. Securing the control protocols while leaving these OAM&P interfaces unprotected opens huge security vulnerabilities. NEs are an attractive target for those who want to disrupt or gain free access to telecommunications facilities, and much has been written about this subject since the 1980s. A magazine, *2600*, devoted to such activities, has been published quarterly for over 25 years. At one time, careful access controls and password management were a sufficient defense, but no longer. Networks using the TCP/IP protocol suite are vulnerable to, among other things, forged source addresses, recording and later replay, packet sniffers picking up passwords, re-routing of traffic to facilitate eavesdropping or tampering, active hijacking attacks on TCP connections, ploys against applications like web services, and a variety of denial of service attacks. In the 1990s, telecommunications facilities were identified in the U.S. as part of the “critical infrastructure,” and increased emphasis was placed on thwarting such attacks from a wider range of well-funded and determined potential adversaries. The ease of forging TCP/IP packets is the main reason network management protocols lacking strong security have not been used to configure NEs (e.g., with the SNMP SET command). Readily available hacking tools exist that let an eavesdropper on a LAN take over one end of any TCP connection, so that the legitimate party is cut off. In addition, enterprises and carriers in some jurisdictions need to safeguard data about their users and network configurations from prying. An attacker could eavesdrop and observe traffic to analyze traffic usage patterns and map a network configuration; an attacker could also gain access to systems and manipulate configuration data or send malicious commands. Therefore, in addition to authenticating the human user (see [T1M1]), more sophisticated protocol security is needed for OAM&P interfaces, especially when they are configured over TCP/IP stacks. Finally, relying on a perimeter defense, such as firewalls, is insufficient protection against “insider attacks,” or penetrations that compromise a system inside the firewall as a launching pad to attack NEs.

### 1.2 Scope

The scope of this IA is to define objectives for securing OAM&P access to NEs and to show how to use different protocol security systems, depending on the OAM&P protocol and security requirements, to achieve these objectives.

The emphasis in this IA is on protocol security between a Management System and NE. This IA does not differentiate strongly among security attributes associated with a human user, process, application, and system. In many cases, there may be no direct human user involved in an operation, and many NEs and OAM&P systems do not distinguish different “user-IDs” or applications. However, in addition to using the protocol security methods in this IA, other methods should be used where available to enforce access controls based on such distinctions.

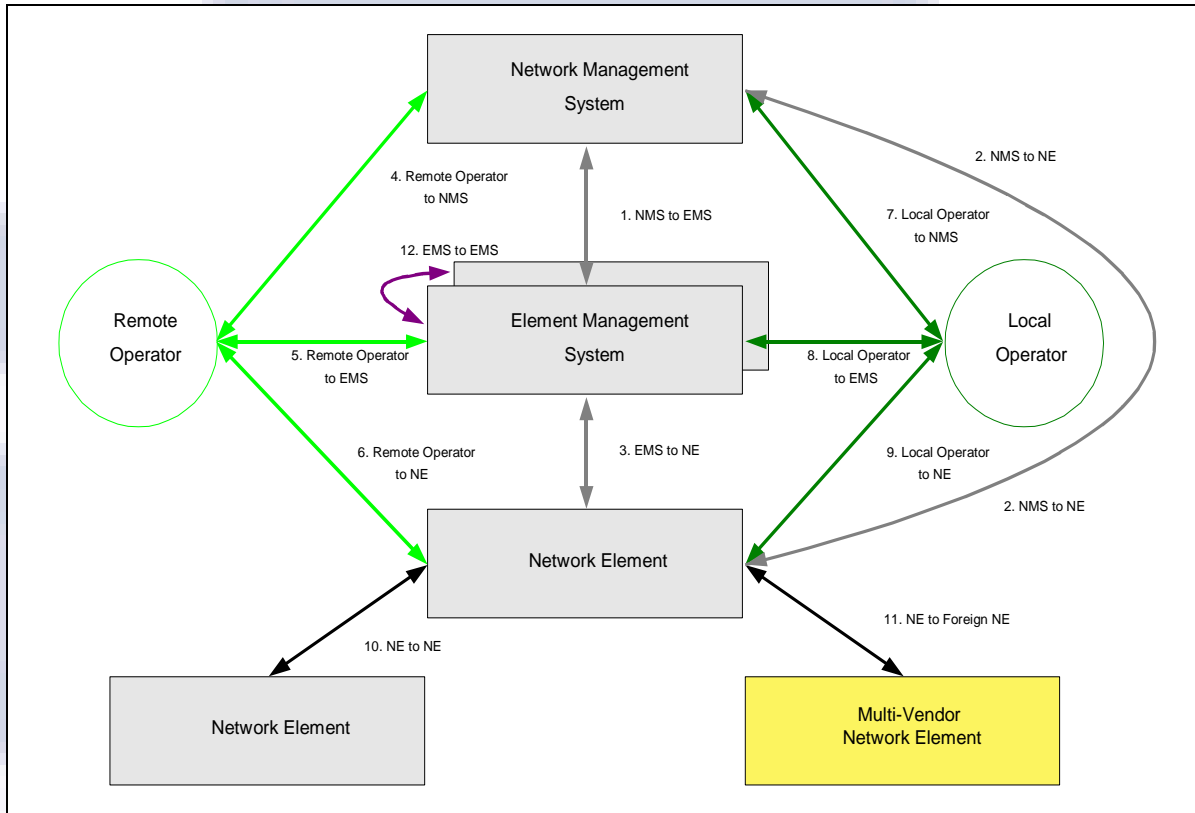
System security of the NEs, Network Management Systems (NMS), and Element Management Systems (EMS) is out of scope, although some remarks in this IA may address the need to safeguard the cryptographic protocol protections themselves. System security for network elements is vitally important and is addressed elsewhere. For more on information assurance requirements, system security requirements, and security-related functional requirements that products can be developed to meet, please refer to the Common Criteria [CC] and appropriate ISO standards [ISO].

The threats identified in the Problem Statement emphasize the vulnerabilities of running OAM&P interfaces over TCP/IP protocol stacks, so this IA addresses protocol security for TCP/IP-based OAM&P interfaces. Other types of OAM&P interfaces exist, from sophisticated ones such as the data communications channels over SONET described in [ANSI95] to simple hard-wired RS-232 connections. In addition, the ITU-T [ITUDCN] has described hybrid networks that use IPv4 as well as other network layer protocols over a variety of layer 2 infrastructures together with encapsulation or tunneling methods. Some of the methods in this IA can possibly be applied to such configurations. For example, protocols tunneled over IPv4 or IPv6 can be protected with IPsec. However, protocols other than those running over TCP/IP stacks are, in general, out of scope for this document.

The rationale for this IA is that NEs using the OIF’s Security Extension for UNI and E-NNI [Sec12] have strong security available for their optical control plane protocols, but security on these NEs may be weakened or compromised by not having comparable protection for their OAM&P interfaces. Therefore, the immediate scope of this IA is NEs that run the OIF’s UNI or E-NNI signaling or routing protocols. However, most, perhaps all, of the material in this IA is not particular to *optical* network elements, but applies to any network element (NE) and its interfaces to Management Systems. These interfaces are shown as number 3 in Figure 1 (or as number 1 for the case in which the EMS and NE are packaged as a single entity). In fact, many optical NEs also switch or route traffic over other types of networks besides optical ones. These NEs are usually managed by a single set of OAM&P protocols running over a set of OAM&P interfaces. In these cases, the security measures described in this IA may be applied to all such OAM&P processes.

NEs, in the larger sense, may be “distributed systems” that are divided, for example, into control and transport components or that have proxy components to carry out certain functions. Multiple components may also exist for enhanced availability or load balancing. Each such component that has its own OAM&P interface(s) is regarded as a NE in this IA. Section 6.5.4.2 of the UNI 2.0 Common Part [UNI2.0] describes different

Service Invocation Configurations and Signaling Transport Configurations. Again, the primary intention is to protect NEs running the OIF’s signaling protocols, but wider applicability is not precluded. For sample descriptions of such configurations, see [ANSI95] and [ITUDCN].



**Figure 1: Network Management Security Reference Model (from [T1M1]).**

### 1.3 Relationship to other Standards Development Organizations (SDOs)

This Implementation Agreement relies entirely on specifications of security developed in other SDOs. It consolidates, profiles, and applies many aspects of the work done in other SDOs to show how different security systems can be used to satisfy management security objectives. The IETF has developed numerous systems for security (e.g., IPsec, TLS, and SSH) and management security protocols (e.g., SNMPv3 and ISMS) that are used in this IA. Section 5.5 discusses work done in additional SDOs that are developing standards for Web Services and Web Services Security, in particular, W3C and OASIS.

Management security has been an ongoing activity in several other SDOs. The ATM Forum published a specification on securely managing ATM network elements [ATMF02]. This IA is patterned after the ATM Forum’s document; certain aspects of the two documents are quite similar, other aspects are substantially different.

The American National Standard T1.276-2003, Baseline Security Requirements for the Management Plane, was originally submitted by Committee T1 to ATIS T1M1.5 and



approved as a standard for providing security requirements to allow for the implementation of secure network management for management systems.

T1M1 has written security requirements for the management plane [T1M1]. Their document addresses security of the management plane for the public switched network, and this document is aligned with the terminology and reference diagram used by T1M1. Note that [T1M1] places requirements on NEs for security, defines administrative roles, places requirements on end systems, and also addresses physical security; this document does not. This document, written to be consistent with the requirements specified in [T1M1], provides narrower guidance for selecting, implementing, and deploying protocol security systems to protect TCP/IP-based management interfaces to NEs.

The International Organization for Standardization (ISO) has developed Guidelines for the Management of IT Security (GMITS). Five separate documents were written on various aspects of managing IT security:

- Part 1: Concepts and models for IT Security
- Part 2: Managing and planning IT Security
- Part 3: Techniques for the management of IT Security
- Part 4: Selection of safeguards
- Part 5: Management guidance on network security

ISO is now replacing these with a larger set of documents in the 27000 series [ISO].

ITU-T SG 15 is the study group for optical and transport network infrastructures. They have produced G.7718 and G.7718.1, ASON Control Plane Management. The ITU-T SG 4 was the lead study group on telecommunication, network, and next generation network management. This study group established the NGN Management Focus Group in September in 2004. They have written a series of Security of the Management Plane, M.3016, documents. The M.3016 series consists of:

- M.3016.0: Overview
- M.3016.1: Requirements
- M.3016.2: Services
- M.3016.3: Mechanisms
- M.3016.4: Profile proforma

The 3GPP SA 5 has a series of documents as well for NGN Management:

- 32.101/32.102: Principles and architecture
- 32.150 series: IRP methodology
- 32.111 series: alarm IRP
- 32.200 series: subset for IMS charging and billing
- 32.300: Common Management Series
- 32.400: Performance Management series

The TeleManagement Forum (TMF) has worked on developing Multi-Technology Network Management (MTNM), but this work has not addressed security.

The IETF has also published standards for formatting and securing log records with syslog. The OIF used these standards to develop an Implementation Agreement for controlling and securing the logging of events at a NE [Syslog12].

This work in the OIF complements these activities by filling in security objectives and showing how a complete protocol security system can be applied to network management.

#### **1.4 Acknowledgements**

The OIF acknowledges the contributions of the ATM Forum and T1M1 mentioned above. The following people contributed to earlier versions of this document:

Gary Buda, Booz Allen Hamilton

Renée Esposito, Booz Allen Hamilton

Richard Graveman, Telcordia

Brian Hazzard, PhotonEx

Jim Jones, Alcatel-Lucent

Scott McNown, Department of Defense

John Naegle, Department of Defense

Dimitrios Pendarakis, Tellium

Tom Tarman, Sandia National Labs

#### **1.5 Outline of the Implementation Agreement**

This Implementation Agreement (IA) consists of two main parts. The first (Section 3) lists objectives for securing the protocols used over OAM&P interfaces to a Network Element (NE). The second (Sections 4–6) presents a model for securing these protocols at different layers, describes systems that are well-suited to secure these interfaces at various protocol layers, lists necessary details for using these security systems appropriately, and summarizes how such security systems achieve the objectives in the first part. The term *objectives* is used in Section 3 because not all of these items are requirements for all users. Users are advised to decide which objectives are requirements for them and to choose a solution described in Section 5 that best meets their requirements. Each security system provides multiple security services, e.g., authentication, integrity, and confidentiality. A major goal of this IA is to define interoperable and high-quality security solutions for these OAM&P interfaces. This is accomplished by showing how to use these security systems simply and effectively to achieve as many of the listed security objectives as possible.

This Implementation Agreement updates and obsoletes *Security for Management Interfaces to Network Elements* (OIF-SMI-01.0) [OIF03] and the *Addendum to the Security for Management Interfaces to Network Elements* (OIF-SMI-02.1) [OIF06].

One main purpose of this IA is to show how to apply security to management interfaces to NEs by using:

- High-quality, standard systems of security protocols, which provide a full range of security services and mechanisms and have multiple interoperating implementations,
- Integrated and automated key management, and
- Consistent identification, authentication, and authorization of network administrators (NAs). Note that [T1M1] identifies different types of administrators with different roles. In this document, the term “NA” applies to any and all of these.

Management interfaces include all access methods and protocols used for network or element management, administration, operations, maintenance, and related tasks.

Terminology and acronyms are presented in Section 2. Then, part one begins in Section 3 by enumerating objectives for securing management interfaces to a NE. The following three sections focus on how to apply existing security systems (e.g., TLS, SSH, SNMP security, or IPsec) to provide secure management access to a NE. Section 4 describes the different types of management interfaces, the protocol stacks they may use, and where the different security systems fit into a typical TCP/IP protocol stack. To promote interoperability, it recommends a preferred solution. Section 5 briefly describes the different security systems, provides references to them, and lists additional details needed for using them appropriately. Section 6 shows the extent to which the proper use of these security systems satisfies the objectives in Section 3. This IA does not define any new protocols or management information.

## **1.6 How to Use this Implementation Agreement**

Vendors of NEs or Management Systems should determine which protocol stacks their OAM&P interfaces use and refer to the appropriate sections for guidance on which security alternatives they have and which options to prefer in each of these cases.

Service Providers and enterprises should first examine the security objectives in Section 3 to determine which security objectives are critical requirements for their operations. Then, they should use this document to map their requirements to the most appropriate security solutions.

## **1.7 Document Organization**

This document is organized as follows:

- Section 2 defines the terminology and acronyms used.
- Section 3 lists and describes the security objectives.
- Section 4 describes the typical protocol stacks used by management interfaces and where security systems fit into these stacks. Among these, it recommends one choice.
- Section 5.1 covers securing protocols that run over IP with IPsec.
- Section 5.2 covers securing protocols that run over TCP with TLS.

- Section 5.3 covers securing MIB-based management systems with SNMPv3.
- Section 5.4 covers securing command line protocols or SNMP with SSH.
- Section 5.5 covers securing web-based management systems.
- Section 5.6 covers use of these solutions together with Radius, S/MIME, or syslog.
- Section 6 maps the security systems in Section 5 to the objectives in Section 3.
- Section 7 contains a summary, and Section 8 contains normative and informative references.

Each subsection of Section 5 presents a general description and guidelines for using one security system aimed at satisfying the security objectives in Section 3. Table 6 in Section 6 summarizes which security objectives from Section 3 are fulfilled by following the guidelines in Section 5. The guidelines for using each of the security systems in Section 5 are aimed:

1. To help systems secured by the given security system satisfy the security objectives in Section 3,
2. To promote interoperability of such implementations with commonly available and current implementations, and
3. To help configure these systems according to generally accepted best practices.

## 2. Terminology and Acronyms

### 2.1 Keywords

When written in ALL CAPITALS, the key words “MUST”, “MUST NOT,” “REQUIRED,” “SHALL,” “SHALL NOT,” “SHOULD,” “SHOULD NOT,” “RECOMMENDED,” “NOT RECOMMENDED,” “MAY,” and “OPTIONAL” in Sections 4 through 6 of this document are to be interpreted as described in IETF RFC 2119 [Bra97].

### 2.2 Terminology

In this implementation agreement, the following definitions apply:

**Network Element (NE):** Any device implementing one or more of the OIF’s UNI or NNI control protocols. It may also support other interfaces or services. In this IA, a networking component with its own OAM&P interfaces (e.g., a signaling controller or transport component), is considered a NE.

**Element Management System (EMS):** A terminal, network element, or system that provides specific services to manage specific Network Elements.

**Network Management System (NMS):** A terminal, network element, or system that provides services to manage a Network Element. It may be an overall management system that manages multiple EMSs and Network Elements, including non-optical Network Elements.

**Management System:** A generic term for an EMS or NMS.

**Network Administrator (NA):** A person who is authorized to use a Management System. (Refer to [T1M1] for the many roles that may exist for a NA.)

### 2.3 Acronyms

The following acronyms or abbreviations are used in this implementation agreement:

<b>AES</b>	Advanced Encryption Standard
<b>AH</b>	Authentication Header
<b>CA</b>	Certification Authority
<b>CBC</b>	Cipher Block Chaining (Mode)
<b>CFB</b>	Cipher Feedback (Mode)
<b>CHAP</b>	Challenge Handshake Authentication Protocol
<b>CORBA</b>	Common Object Request Broker Architecture
<b>CTR</b>	Counter (Mode)
<b>DES</b>	Data Encryption Standard
<b>DH</b>	Diffie-Hellman
<b>DNS</b>	Domain Name System
<b>DSS</b>	Digital Signature Standard
<b>DTLS</b>	Datagram Transport Layer Security
<b>EMS</b>	Element Management System
<b>E-NNI</b>	External Network-Network Interface
<b>ESP</b>	Encapsulating Security Payload
<b>FQDN</b>	Fully-Qualified Domain Name
<b>GCM</b>	Galois Counter Mode
<b>HMAC</b>	Hashed Message Authentication Code
<b>HTML</b>	Hypertext Markup Language
<b>HTTP</b>	Hypertext Transfer Protocol
<b>ICMP</b>	Internet Control Message Protocol
<b>IETF</b>	Internet Engineering Task Force
<b>IKE</b>	Internet Key Exchange
<b>IKEv2</b>	Internet Key Exchange version 2
<b>IP</b>	Internet Protocol version 4 or Internet Protocol version 6
<b>IPsec</b>	IP Security
<b>MAC</b>	Message Authentication Code
<b>MIB</b>	Management Information Base
<b>NA</b>	Network Administrator
<b>NE</b>	Network Element

<b>NMS</b>	Network Management System
<b>NNI</b>	Network Node Interface
<b>NTP</b>	Network Time Protocol
<b>OASIS</b>	Organization for the Advancement of Structured Information Standards
<b>OS</b>	Operating System
<b>PAP</b>	Password Authentication Protocol
<b>PKI</b>	Public Key Infrastructure
<b>PRF</b>	Pseudo-random function
<b>RFC</b>	Request for Comments
<b>RSA</b>	Rivest, Shamir, and Adleman
<b>SA</b>	Security Association
<b>SAD</b>	Security Association Database
<b>SAML</b>	Security Assertion Markup Language
<b>SCTP</b>	Stream Control Transmission Protocol
<b>SHA</b>	Secure Hash Algorithm
<b>S/MIME</b>	Secure Multipurpose Internet Mail Extensions
<b>SNMP</b>	Simple Network Management Protocol
<b>SOAP</b>	Simple Object Access Protocol
<b>SPD</b>	Security Policy Database
<b>SSH</b>	Secure Shell
<b>SSL</b>	Secure Sockets Layer
<b>TCP</b>	Transmission Control Protocol
<b>TL1</b>	Transaction Language 1
<b>TLS</b>	Transport Layer Security
<b>UDDI</b>	Universal Description, Discovery and Integration
<b>UDP</b>	User Datagram Protocol
<b>UNI</b>	User-Network Interface
<b>URI</b>	Uniform Resource Identifier
<b>VPN</b>	Virtual Private Network
<b>W3C</b>	World Wide Web Consortium
<b>WS</b>	Web Services
<b>WSDL</b>	Web Service Definition Language
<b>WSDM</b>	Web Services Distributed Management
<b>WSS</b>	Web Services Security
<b>XACML</b>	Extensible Access Control Markup Language
<b>XCBF</b>	XML Common Biometric Format

<b>X-KISS</b>	XML Key Information Service Specification
<b>XKMS</b>	XML Key Management Specification
<b>X-KRSS</b>	XML Key Registration Service Specification
<b>XML</b>	Extensible Markup Language
<b>XML-DSIG</b>	XML Digital Signature
<b>XML-ENC</b>	XML Encryption

### 3. Threats and Security Objectives

The general threat model is that anyone can read or write arbitrary information on the same network as the legitimate parties. In fact, attacks can be combined: information can be read and modified or deleted, or recorded and played back later in an identical or modified form. Source and destination addresses and other control information (e.g., a TCP reset) and control protocols (e.g., ICMP) can also be forged or manipulated. Anyone can gain full knowledge of the legitimate protocols, including security protocols, being used. However, we assume that no one can completely stop the flow of legitimate packets. Also, the legitimate parties can be initially configured with cryptographic mechanisms and secrets, they can secure their internal state (memory) from reading or tampering, and they can generate cryptographically sound pseudorandom numbers. Providing security to protect against this threat model includes defenses against attacks sometimes labeled as:

- **Masquerade.** Attacks under this heading are often called spoofing, session hijacking, or man-in-the-middle. Masquerade usually implies impersonating the name or address of a legitimate party to gain access, carry out a malicious act, or observe the activities of a system and gain more knowledge about the system's users or configuration.
- **Unauthorized access.** Attacks under this heading include exploiting system vulnerabilities to gain access to and control of system resources, to compromise a network node, to cause incorrect operations, to modify configuration data or software, or to disable security features.
- **Data integrity threats.** This includes modifying, reordering, truncating, or replaying legitimate communications, or outright forgery.
- **Confidentiality threats.** Attacks under this heading include eavesdropping or "packet sniffing," session recording, and disclosure. These attacks may occur when an attacker taps into a transmission facility or network node or otherwise captures data being transferred on a communications channel. An attacker may attempt dictionary attacks to discover passwords or cryptanalysis on captured and encrypted data to recover message properties or contents.
- **Traffic analysis.** This threat consists of an attacker being able to discern the configuration of or usage patterns on a network, including the numbers and types of systems; names of parties; patterns, frequency, and volume of information communicated between them; and the protocol stacks they are using.

- **Denial of Service (DoS).** DoS occurs when an attacker executes commands or performs operations that cause undue loads on communications channels, network nodes, or end systems, which result in resources being unavailable for authorized uses.

Vendors should address the above threats when incorporating security into their products or developing specific security products for their management, administration, operations, and maintenance interfaces between their NEs and Management Systems. For the purposes of this document, “interfaces between NEs and Management Systems” is interpreted broadly to include all OAM&P communications with a NE, regardless of the Management System endpoint. Vendors should consider the following list of security objectives and state which are met by their products. The term *objectives* is used in this section, because not all of these items are *requirements* for all users. Users are advised to decide which objectives are requirements for them and to choose a solution described in Section 5 that best meets the objectives they consider requirements.

### 3.1 Confidentiality

Confidentiality is used to protect data against partial or complete disclosure to unauthorized parties. Information that may need to be protected for confidentiality includes, but is not limited to, statistical data, configuration information, connectivity information, passwords or other security data, and management data transferred between a NE and a Management System. Cryptography can aid in maintaining information confidentiality. It must be noted that providing cryptographic confidentiality also requires entity authentication and data integrity. Users may consider the following objectives for ensuring confidentiality:

- C-1 The interface between the NE and the Management System supports confidentiality of management data transferred between the NE and the Management System.
- C-2 The interface between the NE and the Management System supports confidentiality of passwords and keying material.
- C-3 The interface between the NE and the Management System supports confidentiality of audit information.
- C-4 The interface between the NE and the Management System provides confidentiality of identities and addressing information.

### 3.2 Data Integrity

Data integrity is the ability to ensure that data have not been altered or forged in an unauthorized manner. For example, SNMP messages may have to be protected from being maliciously changed in such a way that the altered message could result in unauthorized management operations, including falsifying the value of an object. Data integrity also ensures that the message sequence has not been altered in a manner that would cause unauthorized management operations. This extends to preventing replay attacks by ensuring that a message is not accepted multiple times or after undue delay.



Note that data integrity cannot be obtained without data origin authentication. Users may consider the following objectives for ensuring data integrity:

- I-1 The interface between the NE and the Management System supports message integrity for communications between the NE and the Management System.
- I-2 The interface between the NE and the Management System supports a mechanism for replay protection for communications between the NE and the Management System.
- I-3 The interface between the NE and the Management System supports integrity of audit information.
- I-4 The interface between the NE and the Management System supports a mechanism to detect delay of communications between the NE and the Management System and prohibits communications that exceed the limits of a time window.

### **3.3 Key Management**

Key management is the supervision and control of the process whereby keys are generated, stored, protected, transferred, loaded, used, and destroyed. Users may consider the following objectives for key management:

- K-1 Based on a secure system of installing pre-shared secrets or public-private key pairs, the interface between the NE and the Management System supports a dynamic key management system for the automated and secure establishment and distribution of key encryption keys (e.g., pre-shared secrets or master keys) that are shared between the Management System and the NE.
- K-2 The interface between the NE and the Management System supports a key management system for the dynamic, automated, and secure establishment and distribution of traffic protection keys (i.e., the keys used to encipher and decipher or to generate and verify integrity checks of the actual OAM&P traffic) that are shared between the Management System and the NE.
- K-3 The interface between the NE and the Management System provides forward secrecy for all confidential communications between the NE and the Management System. Forward secrecy means that subsequent compromise of long-term keys does not also compromise the contents of previous sessions that were set up using these long-term keys. It can be achieved, for example, by using long-term keys only to authenticate, but not to generate or encrypt, traffic protection keys.
- K-4 The interface between the NE and the Management System provides a method for secure rekeying of traffic protection keys. The security of the rekeying is based on the authenticated and shared key encryption keys (K-1).

### **3.4 Authentication**

Authentication protects communicating systems from accepting fraudulent data or revealing data to unauthorized parties by allowing them to verify the identity of the originator or recipient of a message, respectively. (For example, a goal of authentication

may be to verify the identity of the user who claims to have generated a SNMP message.) As described in Section 1.2 on Scope, authentication is defined at the level of a system, but finer grained methods are allowed (e.g., user, process, or application level authentication). Users may consider the following objectives for authentication:

- A-1 The interface between the NE and the Management System supports the capability for each entity to establish and verify the claimed identity of the other.
- A-2 The interface between the NE and the Management System authenticates all messages between the NE and the Management System.

### 3.5 Negotiation and Policy Enforcement

When each OAM&P interface is originally configured, the security policy for using this interface may be specified. In general, stronger security is achieved if NEs and Management Systems are delivered from the vendor with security options enabled and with appropriate warnings about disabling these options. Policy may be enforced by determining the security parameters for a communication session at session establishment. Users may consider the following objectives for negotiation and policy enforcement:

- N-1 The NE is configured to allow specification of the security systems, services, and options it requires for each OAM&P interface.
- N-2 The interface between the NE and the Management System supports secure negotiation of the security services, mechanisms, and algorithms used to protect OAM&P protocols. This may be achieved, for example, if the first party can list acceptable choices for these parameters and the second party can select from these choices which to use. Secure negotiation implies that an active attacker cannot trick the legitimate parties into using a weaker choice (downgrade attack).

### 3.6 Non-Repudiation

Non-repudiation of message origin is the ability to guarantee to a third party the originator's authenticity and the integrity of a message, so that the originator cannot deny having sent the message. Users may consider the following objective for non-repudiation:

- R-1 The interface between the NE and the Management System provides a protocol that supports non-repudiation of message origin.

### 3.7 Access Control

Access control defines and restricts the privilege to access information or perform specific functions to certain entities, roles, or systems. *Entities* are referenced by *user IDs* or *login names* that identify users to different operating systems. *Roles* are defined by *groups* or *privileges granted to entities*, again, depending upon the operating system. In some environments, the role of *Network System Administrator* is distinct from the role of *System Security Administrator*, whereas in other environments a single privileged role is defined. See, for example, [T1M1] for the definition of five types of administrative roles:

Application Administrator, Application Security Administrator, System Administrator, System Security Administrator, and Application User/Operator. The term Network Administrator (NA) is used when referring to all types of administrators. However, it is important to remember that each administrative role may have specific functions and privileges. For instance, a System Security Administrator may be responsible for the proper activation, maintenance, and use of the security features of a system (i.e., NE or Management System). On the other hand, a System Administrator may be responsible for OS-level processes and procedures pertaining to installation, operations, and maintenance of the operating platform; installation of software on the platform; and control of privileged authority. For the most part, the security systems described in Section 5 can function with entities defined at the level of an entire *system*, but most of them may also be used with finer-grained access controls. Thus, users may consider the following objectives for access control:

AC-1 The NE provides the means to limit the actions of a NA based upon the NA's identity or role.

AC-2 The NE provides the means to limit a NA's privileges based on criteria such as the OAM&P port, protocol, time of day, or specific command.

### **3.8 Audit and Event Logging**

Auditing and logging network events provide a chronological record of system activities and allows the examination of sequences of events or changes in state. The information audited and captured in an audit log may be configurable and needs to be protected from unauthorized access, tampering, or removal. Users may consider the following security objectives for auditing and logging network events:

L-1 The NE is capable of recording a set of events that is specified by a NA, according to the access controls granted to the NA.

L-2 The NE is capable of reporting events selected by a NA to the Management System as they occur in real time.

L-3 The NE is capable of recording the system time at which each audited event occurred to a granularity of no greater than one second.

L-4 The NE is capable of recording the identity of the NA who performed each action.

L-5 The NE is capable of presenting the audit data to the NA in such a manner that the data can be interpreted and read from the audit records.

L-6 The NE is capable of detecting and reporting the occurrence of replayed packets.

### **3.9 Denial of Service**

Denial of Service (DoS) attacks can be indistinguishable from certain types of network failures a network management protocol must handle. Preference should be given to security protocols that were designed conscientiously to minimize DoS vulnerabilities. Users may consider the following security objectives for handling denial of service attacks:

- D-1 Safeguards are implemented to ensure that any DoS attack initiated on a NE via a management interface does not affect service to the optical bearer traffic.
- D-2 The NE is capable of gracefully handling known types of DoS attacks.

### 3.10 Traffic Analysis

Traffic analysis consists of determining addresses, types of systems, timing, message counts, protocols, and message lengths. This information can be used to estimate the size, topology, and usage of a network and also to gain information about routing, faults, etc. Users may consider the following security objectives for traffic analysis:

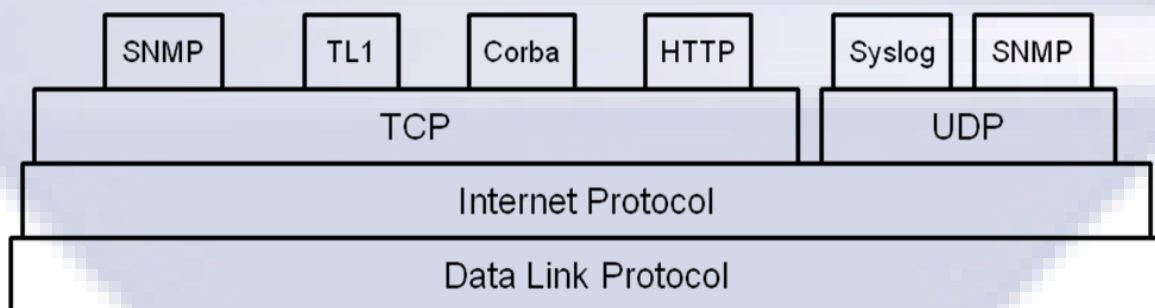
- T-1 The interface between the NE and the Management System protects the confidentiality of parties' identities.
- T-2 The interface between the NE and the Management System supports mechanisms that prevent an eavesdropper from learning network size, topology, or activity from an analysis of message types, lengths, counts, and timing.

## 4. Management Interfaces and Protocol Stacks

The management interfaces described within this IA include:

- Command line interfaces, e.g., telnet or TL1,
- MIB-based management, e.g., SNMP access,
- Any interface running over TCP, e.g., Web access via HTTP or CORBA,
- Any management interface running over IP.

Figure 2 depicts a sample protocol stack that shows protocol options for management systems. For each of these management protocols, Section 5 describes appropriate security systems to provide sufficient protocol security for protection from a wide range of passive and active attacks.

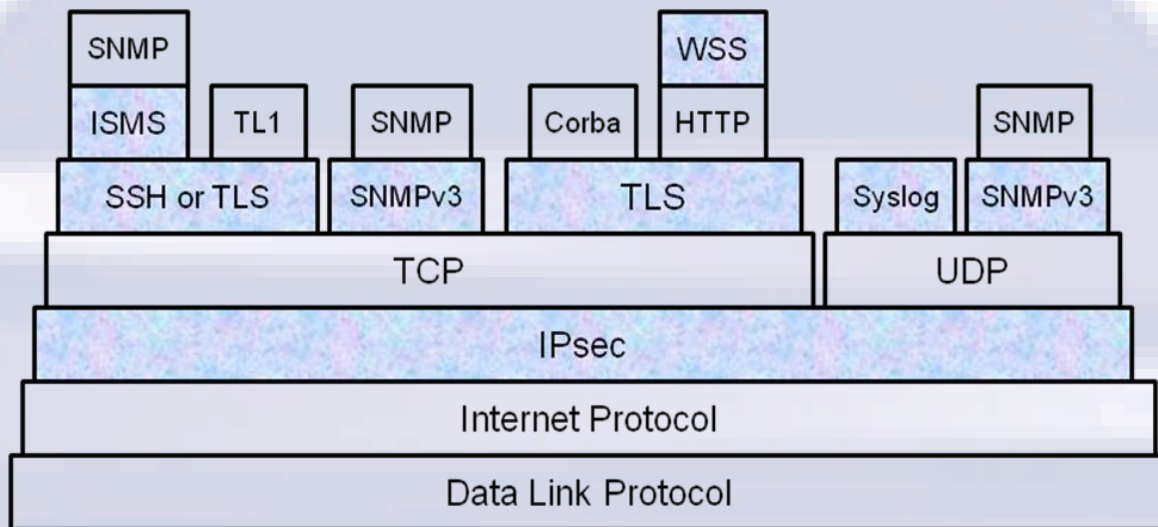


**Figure 2: Typical Protocol Stacks for Management Interfaces.**

### 4.1 Protocol Stacks and Security

Figure 3 is an expanded version of Figure 2 that shows where the security systems of Section 5 (shown in the shaded blocks) can fit into the protocol stack. Command line

interfaces, for example, may be secured in any of three ways: (1) with the Secure Shell (SSH) between the application and TCP layers; (2) with Transport Layer Security (TLS) also between the application and TCP layers; or (3) with IPsec at the IP layer. SNMPv3 is shown as a separate “security envelope” below SNMP (v1 or v2), because it is an application-level security encapsulation of SNMPv1 or SNMPv2. The unshaded blocks represent protocol layers that may contain certain security mechanisms, e.g., CORBA Sec [OMG02] in CORBA, but these mechanisms are either considered insufficient or dependent upon lower-layer mechanisms.



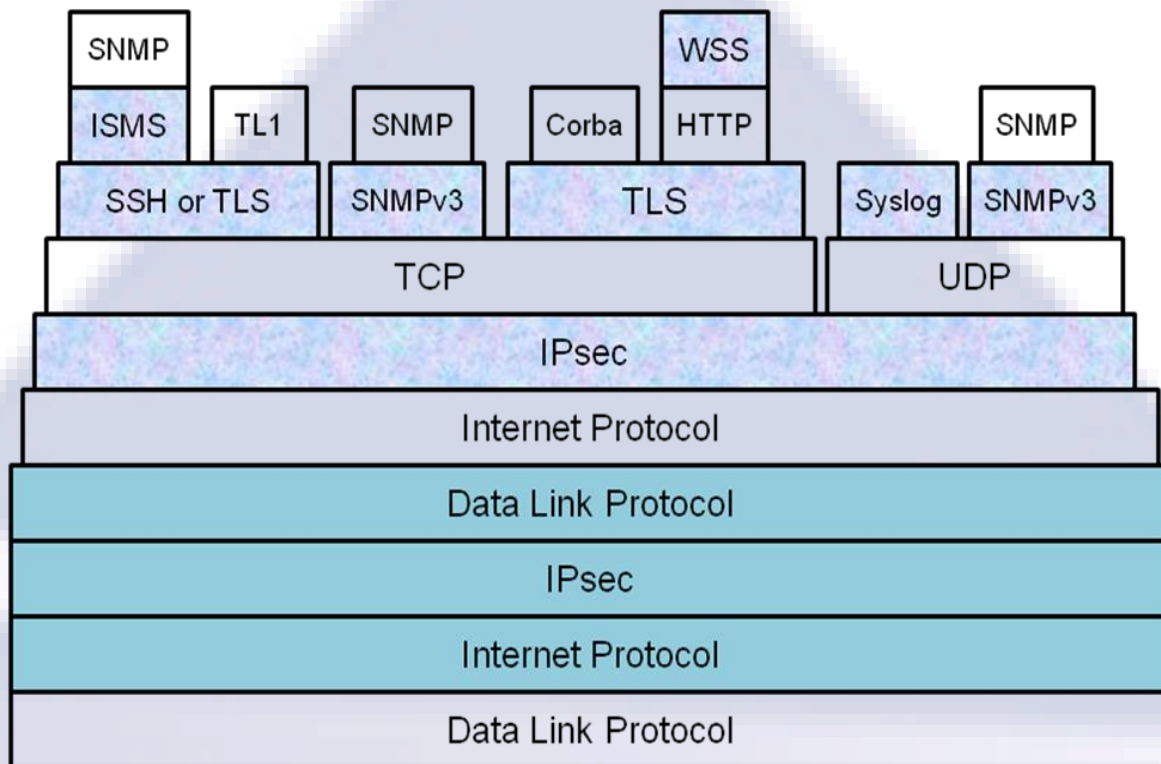
**Figure 3: Protocol Stacks and Security.**

The intent of this IA is to offer a choice of acceptable security systems and to show how to use each appropriately to achieve security between a Management System and a NE. The OIF has defined control plane security based on IPsec [Sec12], and IPsec can run below all of the transport protocols and applications shown, so, in the absence of another adequate solution, IPsec is RECOMMENDED.

## 4.2 Protocol Stacks and VPNs

A remote access connection can use any of the interface types (command line, web, or MIB-based) described above. VPN encapsulation offers an additional choice as to where security can be placed in the protocol stack.

IPsec can protect all traffic across a VPN, IP-based or otherwise, as shown in Figure 4. The lower IP and IPsec layers in Figure 4 (with the darker shading) depict a VPN running over a potentially unprotected network segment. (Above these layers, an emulated link layer may exist, but this is immaterial to the security discussion here.) VPNs operate between routers, firewalls, or security gateways, and, therefore, they do not provide end-to-end security, so end-to-end security may be applied in the upper layers of Figure 4 as well.



**Figure 4: Protocol Stacks Including a Layer 3 VPN.**

Security systems usually provide more than one service (e.g., authentication and integrity) and, as depicted above, they may be combined (see Section 5.6) to provide greater levels of security or alternative methods of authentication.

### 4.3 Management Interfaces and Security Protocols

The security systems shown in Figure 3, Figure 4, and , but this was updated in RFC 6434 [JLN11] to state that IPsec ESP and IKEv2 SHOULD be implemented. This is still a strong recommendation, and it clears up the ambiguity about IKE in RFC 2460.

Table 1 can be applied to different protocols, protocol layers, and types of OAM&P interfaces. The desired security scope and security services needed may influence which security systems are chosen. For example, SSH and TLS normally protect traffic from the Management System to the NE, that is, end to end. IPsec can be implemented from an end-host or a security gateway to another end-host or security gateway. As depicted in Figure 3 and Figure 4, IPsec can be applied to any management interface running over IP. Therefore, the following recommendations are made:

If a NE provides command line access, it MUST support at least one of the following:

- SSH
- TLS
- Lower layer protection with IPsec

If IP is part of the protocol stack, IPsec is RECOMMENDED, and SSH, and TLS are OPTIONAL.

Security is a requirement for all command line interfaces to the NE, regardless of whatever lower layer protocols they may be using. This includes management, administration, debugging, and remote maintenance ports, and any other such interfaces not explicitly listed here. Therefore, any interfaces that do not support one of the above solutions MUST be physically secured or disabled.

If a NE provides MIB-based management access, it MUST support at least one of the following:

- SNMPv3 (with or without an underlying TCP or IP layer)
- ISMS
- IPsec
- TLS (if running over TCP)

If IP is part of the protocol stack, IPsec is RECOMMENDED, and SNMPv3, ISMS, and TLS are OPTIONAL.

If the OAM&P protocols are running over TCP but are not covered by the above cases (e.g., Web-based management with HTTP or CORBA management) they MAY be protected with any of the Web Services Security measures described in Section 5.5, but they MUST also be protected by one of the following:

- TLS
- IPsec

IPsec is the RECOMMENDED choice, and TLS is OPTIONAL.

To summarize, , but this was updated in RFC 6434 [JLN11] to state that IPsec ESP and IKEv2 SHOULD be implemented. This is still a strong recommendation, and it clears up the ambiguity about IKE in RFC 2460.

Table 1 shows the variety of choices for protecting various management interfaces. A ‘√’ indicates that the given protocol can be used to protect the specified interface. Because IPsec can be used in all of the identified cases and is the OIF’s choice for securing signaling protocols between NEs [Sec12], IPsec is the RECOMMENDED solution. This choice is consistent with the fact that IPsec is RECOMMENDED in IPv6. It was originally REQUIRED (see RFC 2460, [DH98]), but this was updated in RFC 6434 [JLN11] to state that IPsec ESP and IKEv2 SHOULD be implemented. This is still a strong recommendation, and it clears up the ambiguity about IKE in RFC 2460.

**Table 1: Applicability of Security Solutions to Different Interfaces.**

<u>Interface</u>	SNMPv3	TLS	SSH/ ISMS	IPsec

Web or CORBA		√		√
MIB based over TCP	√	√	√	√
MIB based over UDP	√			√
Command Line		√	√	√

## 5. Security Systems

### 5.1 IPsec

#### 5.1.1 IPsec Description

For an overview of all IPsec standards, see [FK11]. The architecture of IPsec is defined in [KS05]. IPsec provides cryptographic security for protocols running over IPv4 or IPv6 with the Encapsulating Security Payload (ESP) [Ken05a]), the Authentication Header (AH) [Ken05b]), and cryptographic key management (IKEv2) [KHNE10], which provide different security services. The AH transform protects IP datagrams by providing message integrity and data source authentication with message authentication codes (MACs) and optional replay detection with sequence numbers. ESP provides not only the services of AH but also confidentiality with encryption. In practice, AH is rarely used.

Once an IPsec security association (SA) is established, datagrams can be sent and received securely. A SA, described by an entry in the security association database (SAD), specifies the security services used to protect the traffic carried within the SA. SAs are identified by <SPI, destination address>, where “SPI” is a 32-bit number standing for Security Parameters Index. IPsec determines whether to apply a SA to outbound traffic and what SAs to require for inbound traffic by consulting the entries, called selectors, in the security policy database (SPD).

The parameters for an IPsec SA are typically established by a key management protocol<sup>1</sup>. These parameters include the encapsulation Mode (Tunnel Mode or Transport Mode), algorithms and modes of operation [NISTmodes], session keys, SPI value, and SA lifetime. IKEv2 is an entity authentication and key management protocol. It supports AH and ESP by establishing and managing the SAs in the SAD. IKEv2 sets up an internal bidirectional SA used to protect IKEv2 exchanges. It then creates unidirectional IPsec SAs and their associated parameters and keys. IKEv2 allows use of a flexible suite of public key and private key algorithms and has a number of attractive security features including forward secrecy, anonymity against eavesdroppers, and some protection against denial of service attacks.

IPsec may operate in Transport Mode or Tunnel Mode. When IPsec is used with IPv4<sup>2</sup>, the protocol field in the IPv4 header contains the value for “ESP” or “AH.” In Transport

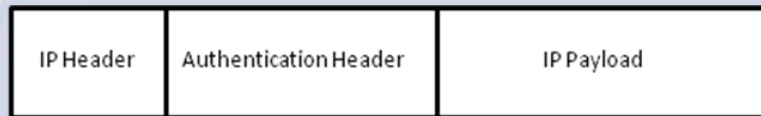
<sup>1</sup> IPsec has a mandatory provision for manual key distribution, but because manual key distribution does not allow for important functions like replay detection and automatic rekeying, it is not recommended in this IA.

<sup>2</sup> With IPv6, ESP and AH are implemented as extension headers.

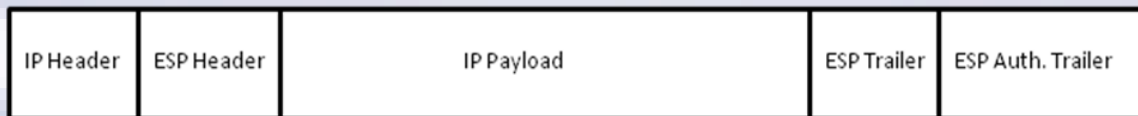


Mode, the next header field in AH or ESP contains the value that was in the original IPv4 protocol field before IPsec processing was applied, e.g., ICMP, TCP, or UDP. The structure of the packet is depicted below in Figure 5.

In Tunnel Mode, IPsec protects an entire IP datagram, and the next header field in AH or ESP contains “IPv4” again. This is depicted below in Figure 6. Transport Mode imposes less overhead, but it is usually used end to end, not at routers or firewalls.

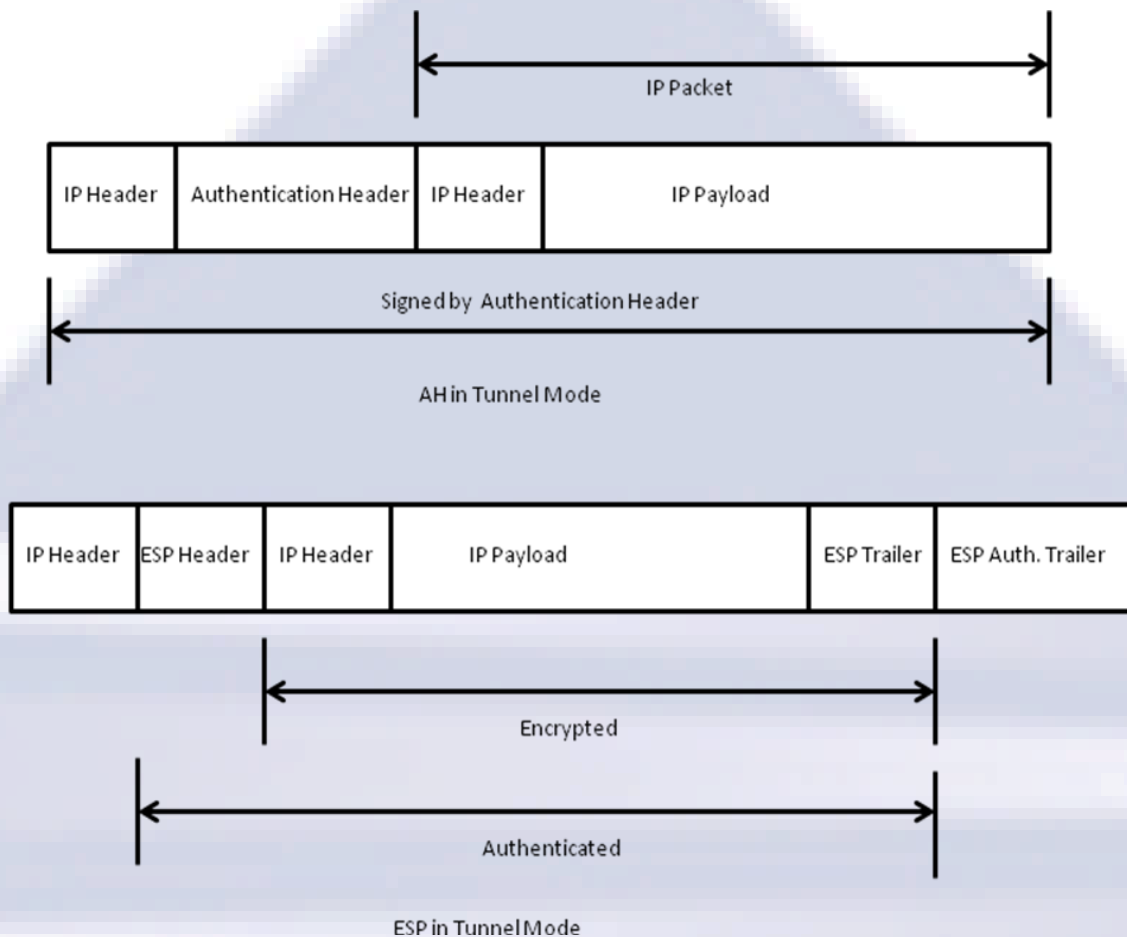


AH in Transport Mode



ESP in Transport Mode

**Figure 5: AH and ESP in Transport Mode.**



**Figure 6: AH and ESP in Tunnel Mode.**

Authentication of the parties using IPsec is implied by the possession of the integrity and confidentiality keys used by AH or ESP. Therefore, entity authentication is tightly linked to the key management protocol during SA establishment. Typically, certificates are used to verify digital signatures or to complete other public-key operations applied within the key management, and entity authentication is achieved by examining the issuer, subject name, and other pertinent information in such certificates, chains of certificates, associated revocation lists, etc. Alternatively, entity authentication may follow from the use of IKEv2 with pre-shared keys. Pre-shared keys require that the key value be administratively configured into each such pair of peers in a secure manner.

The strength of cryptography depends on the algorithms and key sizes chosen. Single DES has been broken and **MUST NOT** be used. The confidentiality algorithms for ESP and IKEv2 are shifting towards the use of AES due to the increased security levels and performance offered. AES can provide different security services (integrity or confidentiality), can be used in different modes, and is defined with three different key sizes.

ESP and IKEv2 may also use hash functions for message integrity checks or signatures. MD5, previously the most widely used hash function, has been broken and **MUST NOT** be used. Attacks are also known against SHA-1, so stronger alternatives **MUST** be included.

AES is also defined with modes of operation that offer a combination of confidentiality and integrity. For instance, AES in Counter Mode (CTR) [Hou04] is recommended as the preferred encryption method for high-speed implementations. However, Counter Mode does not provide data origin authentication and data integrity. AES in Galois/Counter Mode (GCM), AES-GCM-ESP [VM05], combines AES-CTR mode with a secure integrity mechanism. It is suitable for implementation at speeds of 10 Gb/s and higher in hardware. Such hardware implementations can flexibly support any of the AES key sizes.

### 5.1.2 Guidelines for Using IPsec

- The general requirements for using IPsec are in [Sec12].
- The cryptographic methods for ESP are in [Sec12].
- The cryptographic methods for IKEv2 are in [Sec12].
- (Informative) Guidelines for using obsolete versions of IPsec and IKE are in Appendix E of [Sec12].

## 5.2 Transport Layer Security (TLS)

### 5.2.1 Description of TLS

The Transport Layer Security (TLS) protocol provides cryptographic authentication, data stream integrity, and data stream confidentiality for TCP connections. The older SSL (Secure Sockets Layer) protocol is considered obsolete and **SHOULD NOT** be used. TLS is particularly well suited for protecting http traffic between web browsers and servers, but it may be used to protect any protocol running over TCP (e.g., telnet, rlogin, syslog, or SNMP). For background information about TLS, see [Res01].

Three versions of TLS have been standardized: TLS 1.0 [DA99], TLS 1.1 [DR06], and TLS 1.2 [DR08]. Implementations **MUST** support at least one of these versions and **MAY** support more than one.

A serious flaw in the TLS renegotiation mechanism was found in 2009. The exact exploits depend on details in the method of authentication and upper layer protocol. To prevent attacks based on this flaw, both clients and servers **MUST** support the renegotiation indication extension as described in [RRDO10].

### 5.2.2 Guidelines for Using TLS

In typical e-commerce applications, the initial burden of authentication is placed on the server, because the browser can supply the required payment credentials like credit card data when needed. For applications like network management, initial authentication of both parties is critical. One method is to outfit both parties with certificates signed by the

network operator's designated CA, install that CA's root certificate in the clients and servers, and remove all other trusted root certificates from the clients and servers. That is, both parties (when using RSA, for example) respond to the CertificateRequest message with a Certificate message and a CertificateVerify message. A simpler and acceptable alternative method of client authentication is to use a hardware-token-based one-time password system over every new, secured connection. Simple passwords sent over the secure connection may be vulnerable to a number of practical attacks, so these should be used only with carefully constructed constraints (complexity, logging, protection against dictionary attacks, etc.).

- A NE or Management System that provides an HTTP server protected by TLS SHOULD support TLS 1.2 [DR08] and MAY support TLS 1.0 [DA99] or TLS 1.1 [DR06]. (TLS has also been extended in [Cho02], [B-W06], and [Eas10].) Older protocols (e.g., SSLv2, SSLv3, PCT, and S-HTTP) have not been kept up to date and are not covered in this document.
- All implementations, both clients and servers, MUST support the renegotiation indication extension as described in [RRDO10].
- All server certificates MUST comply with PKIX (X.509v3) [Coo08]. Servers SHOULD be identified by a FQDN specified as a dNSName in the subjectAltName field and follow the guidelines in [SH11].
- Clients (e.g., browsers) MAY use certificates to authenticate to the server. They MAY, however, use a token-based authentication system or passwords sent over the protected channel.
- Client certificates (if applicable) and server certificates SHOULD be generated with a lifetime of no more than two years. Entire certificate chains MUST be checked for correct names and valid expiration dates. They SHOULD be checked for revocation.
- Both parties MUST have access to a source of cryptographically strong random or pseudo-random numbers. See [ESC05], [Gut98], [KSF99], and [Koç09] for additional guidelines and recommendations.
- The server MUST support RSA; it MAY support DH-DSS; it MAY support the Kerberos cipher suites described in [MH99]; and it MAY support the Fortezza cipher suites, but see [Res01] for a discussion of limitations using Fortezza as described in [FCK96]. For RSA or DH-DSS, key lengths MUST be at least 1024 bits and both servers and browsers SHOULD support longer keys for these algorithms, up to at least 2048 bits. The same is REQUIRED for all certificates in the chain. Applications requiring confidentiality SHOULD use 3-DES or AES-128. RC4-128 MAY also be supported. Proprietary cipher suites MAY also be used.
- Both parties MUST provide long-term protection for the privacy of their authentication data and the integrity of root public keys they rely upon to verify certificates. Hardware tamper resistance (e.g., a smart card or cryptographic module) is preferable to disk storage, but if disk storage is used, these items SHOULD be

encrypted and password protected, and the system **SHOULD** log all attempted accesses securely.

- Both parties **MUST** protect pre-master secrets, master secrets, and session keys for the duration of their use and destroy them directly thereafter. Use of software that allows unrestricted access to main memory, memory dumps, examination of paging devices, and so forth **MUST** be restricted accordingly. Processes **SHOULD** be locked in main memory and not paged wherever practical.
- Session resumption with a timeout **MAY** be used. The **RECOMMENDED** timeout interval is ten minutes.

#### 5.2.2.1 *Guidelines for Using TLS 1.0*

- Implementations **SHOULD** support TLS 1.2 but **MAY** also support TLS 1.0 for backward compatibility. A NE that supports the TLS 1.0 protocol (the protocol version is major=3, minor=1) **MUST** support it as defined in [DA99] and, optionally, as extended in [B-W06].
- If TLS 1.0 is supported, the requirements for connection closure, use of port numbers, checking the server's identity, and checking the client's identity in [Res00] **MUST** be followed.
- If TLS 1.0 is supported, the name matching rules specified in [Coo08] **MUST** be followed.
- Servers **SHOULD** and clients **MAY** support the use of port numbers as described in [KL00].
- Implementations of TLS 1.0 **MUST** support the AES cipher suites in [Cho02].
- Implementations of TLS 1.0 **SHOULD** support the pre-shared secret mechanisms in [ET05].
- Implementations of TLS 1.0 **MAY** support the compression mechanism in [Hol04].
- Implementations of TLS 1.0 **MAY** support the elliptic curve cipher suites in [B-W06].

#### 5.2.2.2 *Guidelines for Using TLS 1.1*

Implementations **MAY** support the TLS 1.1 protocol (RFC 4346, [DR06]). This version mitigates attacks against CBC-mode encryption found in TLS 1.0 and fixes a bug whereby premature closes made session resumption impossible. It also includes improved documentation, IANA registries, and descriptions of attacks. TLS 1.1 is version Major=3, Minor=2. With TLS 1.1, implementation of AES is **REQUIRED** and use of RC4 is **OPTIONAL**. This applies also to the cipher suites used with pre-shared secrets [ET05] and other extensions.

- Implementations of TLS 1.1 MAY support the secure remote password mechanism in [TWMP07].
- Implementations of TLS 1.1 MAY support the elliptic curve cipher suites in [B-W06].
- Implementations of TLS 1.1 MAY support the extensions in [B-W03].

#### 5.2.2.3 *Guidelines for Using TLS 1.2*

The TLS 1.2 protocol, the most recent version, is described in [DR08]. Implementations MAY support TLS 1.0 or TLS 1.1 for backward compatibility. Changes in TLS 1.2 include:

- Basing all PRFs on SHA-2 and dropping the combined use of MD5 and SHA-1.
- Enhancing both parties' ability to specify acceptable cryptographic methods.
- Folding in AES directly and supporting combined mode algorithms.
- Tightening up several requirements including checking EncryptedPreMasterSecret version numbers.
- Allowing variable length Verify\_data.
- Requiring Alerts in many cases and handling RSA padding attacks cleaner.
- Handling empty certificate lists.
- Making TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA mandatory to implement.
- Adding HMAC-SHA-2. Removing DES and IDEA.
- Removing the requirement for backward compatibility with SSLv2.
- Adding advice for implementers.

For TLS 1.2:

- Implementations SHOULD include the AES-GCM cipher suites in [SCM08] and the AES-GCM, SHA-2, elliptic curve suites in [Res08].
- Implementations MAY also support the AES-GCM, SHA-2 cipher suites with pre-shared keys as described in [Bad09] and the accompanying elliptic curve cipher suites described in [BH09].
- Implementations MAY support the extensions in [Eas10].

#### 5.2.2.4 *Securing the Browser*

This section applies to the client software (i.e., browsers) used with TLS to protect HTTP-based OAM&P access to a NE.

- Up-to-date browsers MUST be used instead of older ones, because older protocols like SSLv2 and SSLv3 have security defects, cryptographic strength has increased

since the easing of U.S. export restrictions in January 2000, and other security patches and improvements appear continually. Older, U.S. export-only versions **MUST NOT** be used.

- The browser **SHOULD** be configured so that its security settings support the guidelines listed above. If features such as plug-ins, Java, JavaScript, ASP, or ActiveX controls are not used, they **SHOULD** be disabled. If such features are used, their potential vulnerabilities **SHOULD** be understood and mitigated. Unneeded CAs' certificates **SHOULD** be removed. The browser and the platform on which it is running **SHOULD** be isolated from the possibility of unauthorized modification. Extraneous network services **SHOULD** be disabled. System logging and intrusion detection tools **SHOULD** be used to monitor the configuration as appropriate.
- The browser **SHOULD** wait for the server's handshake Finish message before sending application data.

### 5.3 SNMP

SNMPv1 and SNMPv2 offer limited security, and, therefore, SNMPv3 was specified to provide encryption and authentication as part of the core protocol. SNMPv3 with the user based security model recognizes three levels of security:

1. Without authentication and without privacy (noAuthNoPriv)
2. With authentication but without privacy (authNoPriv)
3. With authentication and privacy (authPriv)

This section describes security for an interface between a NE and Management System that uses MIB-based network management running SNMPv3. Section 5.4.3 describes the more recent ISMS system for securing SNMP with SSH, and Section 5.4.4 provides guidelines for its use.

#### 5.3.1 SNMP over Different Transport Layers

A NE that supports SNMP access over TCP **MUST** support one of the following:

- ISMS as described in Section 5.4.3 and Section 5.4.4,
- SNMPv3 as described below in Section 5.3.2,
- TLS as described in Section 5.2,
- IPsec as described in Section 5.1, which, in this case, is **RECOMMENDED**.

A NE that supports SNMP access over UDP **MUST** support one of the following:

- SNMPv3 as described below in Section 5.3,
- IPsec as described in Section 5.1, which, in this case, is **RECOMMENDED**.

A NE that supports SNMP access over protocols other than TCP and UDP **MUST** support:

- SNMPv3 as described below in Section 5.3.

### 5.3.2 SNMPv3 Description

SNMPv3 is defined in [HPW02], [CHPW02], [FLRW03], [LMS02], [BW02], and [WPM02]. It provides for message integrity, confidentiality, a freshness window, and a strong model for authorization and access control. Parties are authenticated by the possession of shared keys. The SNMPv3 specification names DES-CBC as the only confidentiality algorithm, but newer alternatives (e.g., 3-DES and AES) have been proposed and SHOULD be implemented. For message authentication and data integrity, the SNMPv3 specification lists HMAC-MD5-96 as “shall support” and HMAC-SHA-96 as “should support.” The former MUST NOT be used. SNMPv3 provides a timeliness feature only if authentication is used. The complete SNMP message is checked for integrity, so in conjunction with authentication the timeliness values will be considered trustworthy. SNMPv3 specifies a time window of 150 seconds within which SNMP messages shall be received after the time they are sent. To avoid delay and replay attacks, messages without recent time indicators are not considered authentic. The time of the SNMP engine is indicated by two values taken together, snmpEngineBoots and snmpEngineTime. These two values are included in an authenticated message sent to or received from a SNMP engine. Upon receipt, the values are checked to ensure that the indicated timeliness value is within the acceptable time window.

Again, as with synchronization, timeliness checking is done only if the authentication service is in use and the message is authentic, thus assuring the validity of the message header fields.

Many SNMP implementations make use of proxy agents. SNMPv3 specifies that a proxy forwarding application, “must perform a translation of incoming management target information into outgoing management target information. How this translation is performed is implementation specific.” This implies that proxy agents shall have access to the SNMP packets. Therefore, the proxy agents need to have access to privacy keys and authentication keys. A secured path between a Management System and a NE may include several proxies processing plaintext messages in the path. In fact, any proxy agent in the path may translate a secure message into an insecure message.

SNMPv3 contains no provision for security association negotiation or session key generation. Although SNMPv3 does provide guidelines for the creation, update, and management of the keys, the keys are not accessible via SNMP. SNMPv3 assumes that the user will select the proper key to use for each service and will somehow have distributed the key in a secure manner to all SNMP engines that require it.

SNMP does not contain any specific measures with respect to DoS attacks.

### 5.3.3 Guidelines for Securing MIB-Based Management with SNMPv3

- Entities implementing the rekey option MUST have access to a source of cryptographically strong random or pseudo-random numbers. See [ESC05], [Gut98], [KSF99], and [Koç09] for additional guidelines and recommendations.



- The key localization algorithm transforms the user's password into a traffic encryption key shared between a user and one authoritative SNMP engine. Implementations of SNMPv3 using an auxiliary key management scheme like IKEv2 MUST NOT use the key localization algorithm option.
- SNMPv3 implementations using the integrity option SHOULD use the timeliness feature. If another accurate and secure source of time is not available, NTP (see [Mil10] and [HM10]) is RECOMMENDED.
- Access control lists MAY be used to restrict the IP addresses from which different SNMP messages are sent.
- SNMP agent logging MAY be enabled.
- An SNMP engine MUST discard SNMP Response messages that do not correspond to a Request message.
- Confidentiality, when used, MUST be applied to SNMP packets as described in [Blu04]. [Blu04] specifies the use of AES in Cipher Feedback Mode (CFB) with a key size of 128 bits. DES-CBC MUST NOT be used.
- When confidentiality is used, the accompanying authentication protocol SHOULD be HMAC-SHA-96. Use of HMAC-MD5-96 should be phased out as quickly as practical.

## 5.4 Secure Shell and Integrated Security for Management Systems

### 5.4.1 SSH Description

The Secure Shell (SSH<sup>3</sup>) defines security protocols that use public key cryptography to establish secure, authenticated sessions between a client and a server.

SSH1 [YI96] and SSH2 [BSB05] are two completely distinct protocols. Both have freely available specifications and have been implemented in freeware and commercial products. SSH1 was never standardized, and security weaknesses in SSH1 have been published. SSH2 contains improvements in performance, security, and portability over SSH1. In particular, certain active attacks against the SSH1 protocol are prevented in SSH2. Therefore, SSH1 is considered obsolete. SSH2 has been approved as Proposed Standard by the IETF and is described in five RFCs ([LL06], [YL06a], [YL06b] updated for SHA-2 in [BB12], [YL06c], and [YL06d]).

Note that because SSH1 and SSH2 servers bind to the same TCP port, and the protocol begins with an exchange of protocol and software version numbers, it is possible for a SSH2 server to provide a backward-compatible bridge to handle a SSH1 client.

SSH2 is intended to allow a user to logon, execute commands, or transfer files securely. It is a replacement for telnet, rlogin, rsh, and rcp. It provides strong authentication and secure communications. An integrated "port forwarding" feature can be used to secure

---

<sup>3</sup> SSH is a registered trademark and Secure Shell is a trademark of SSH Communications Security Ltd. of Finland.

X11 connections or in fact any TCP connection, e.g., to perform a secure remote backup. SSH2 has an explicit capability to secure ftp (which has both a control channel and a data channel) as well.

A description of SSH2 begins with the transport layer protocol. A client sends an authentication request to a server, and the server responds with its long-term public host key. The client compares the host key with what has already been configured. A client may be configured to trust new host keys or not. Note that certificates are not used by SSH. To make sure that these first two messages of the key exchange sequence itself have not been manipulated, both parties compute a hash of the initial messages and session key, which they use later as a session identifier.

After the client receives and verifies the server's public key, it chooses a 256-bit pseudorandom number, which becomes the basic shared secret from which all unidirectional session keys are derived. This random number, a known constant, and the session identifier are encrypted with the host key. This value is returned along with a choice of traffic protection algorithms.

SSH2 provides for the negotiation of both traffic protection and compression algorithms. SHA-1 and 3-DES are mandatory to implement, but other popular choices as well as proprietary algorithms can also be used. A reliable transport stream in each direction (i.e., TCP) is required, and packet sequencing is additionally verified by including an implicit sequence number in each MAC calculation. Either party may request rekeying at any time.

The SSH2 authentication protocol is layered on top of the transport layer protocol. The next step is user authentication, which can be done with a password over the secure channel, token-based systems, or the user's public-private key pair. In the last of these cases, a pass-phrase protects the user's private key on the client's system. After the authentication protocol completes successfully, the client may request different protected services from a list of supported services. These services are then protected with SSH2 encryption, MACs, and secured end of file messages.

#### 5.4.2 Guidelines for Using SSH

Note: An OAM&P interface between a NE and a Management System MAY be secured by running an application-layer protocol such as secure telnet, SFTP (Secure FTP), or SCP (Secure Copy) on top of SSH, so long as the underlying SSH2 layer follows the guidelines below.

Because of the improved security of SSH2, the widespread availability of client and server implementations of SSH2, and the standards status of SSH2, use of SSH1 is deprecated. Management interfaces protected with SSH MUST use SSH2 and MUST NOT use SSH1.

Implementations SHOULD include Generic Message Exchange Authentication for SSH2 [CF06] and Diffie-Hellman Group Exchange for the Transport Protocol [FPS06]. Implementations MAY support RSA key exchange as described in [Har06] and the elliptic curve methods as described in [SG09].

Implementations MAY use DNS to publish SSH2 key fingerprints [SG06] and include the Session Channel BREAK in [GL06], the PKI extensions in [GT06] and [GVB07], X.509 certificates as described in [IS11], and the rekeying in [BKN06]. Implementation of the GSS-API extensions [HSGW05] is also OPTIONAL.

Implementations MUST support 128-bit AES in CBC mode. Implementations MAY support 3-DES in CBC mode and AES-GCM as specified in [IS09]. Implementations MUST NOT support DES.

Implementations MUST support SHA-1 and MUST not support MD5. SHA-2 as specified in [Igo11] SHOULD also be supported.

Official releases of the software from SSH Communication Security are signed. Implementers or users downloading these releases of SSH SHOULD verify these signatures.

Implementations of SSH clients and servers MUST use a cryptographically strong method of generating pseudo-random numbers. See [ESC05], [Gut98], [KSF99], and [Koç09] for additional guidelines and recommendations.

Deployments of SSH SHOULD use public key authentication. The public key MAY be that of a specific user's account or the NE. Deployments MAY also use passwords. Host-based authentication SHOULD NOT be used.

Client computers MUST be protected from attempts to modify their configured host keys or to obtain their private keys. Such protection includes physical access to and modification of the software, as well as other compromises.

Clients MUST NOT accept new, not configured host keys for access to NEs.

SSH servers MUST be protected so that host private keys are not revealed, and, in the case of public key authentication, users' public keys are not altered. If passwords or another type of authentication is used, such authentication data MUST also be protected appropriately to avoid both direct attacks and dictionary attacks.

SSH SHOULD NOT be configured with RSA or Diffie-Hellman public key sizes shorter than 1024 bits nor with elliptic curve public key sizes shorter than 160 bits.

If a NE runs a SSH server, it MAY be configured with a SSH client as well.

In UNIX-based implementations, the server (sshd) SHOULD be run directly and not from inetd. It MAY be configured with TCP Wrappers.

### 5.4.3 ISMS Description

SNMP has been widely and successfully deployed, but the lack of strong security has always limited its usefulness. After lengthy discussions and a couple of false starts, the IETF defined SNMPv3, a satisfactory security standard. When it became obvious that SNMPv3 was not being used significantly, the IETF formed the Integrated Security Model for SNMP (ISMS) Working Group. This working group published four RFCs defining a method to secure SNMP with SSH2:

- RFC 5591, Transport Security Model for the Simple Network Management Protocol (SNMP) [HH09], describes the transport security model for SNMP and includes a portion of the MIB for monitoring it. It provides a framework and message processing rules for communicating a securityName (SNMP principal) and securityLevel (security services required or provided) between the SNMP application and security system.
- RFC 5590, Transport Subsystem for the Simple Network Management Protocol (SNMP) [HS09], defines a new component of the SNMP Engine (see [HPW02]) called the Transport Subsystem, which includes security parameters. It defines an abstract services interface, which allows different transport and security systems to operate as a layer underneath the SNMP application. It supports an access control model that separates authentication (message integrity) from authorization, and it provides compatibility with a security system that has a notion of sessions. This requires a longer-term notion of state information, beyond that of a request-response.
- RFC 5592, Secure Shell Transport Model for the Simple Network Management Protocol (SNMP) [HSH09], defines a particular transport model for secure SNMP based on SSH2. It also defines the MIB objects for monitoring and managing the SSH transport model. The SSH transport model supports all of the user authentication methods available in SSH2.
- RFC 5608, Remote Authentication Dial-In User Service (RADIUS) Usage for Simple Network Management Protocol (SNMP) Transport Models [NN09], defines a general method for using RADIUS to authenticate and authorize an SNMP user and illustrates how this can be used with the SSH transport model.
- The main motivation for choosing SSH was to unify the security systems for SNMP and command-line interfaces used in network management. More recently, it has been suggested that TLS (over TCP) or DTLS (over SCTP or UDP) may offer a better operational and security transport model for SNMP than SSH2 does, and a Proposed Standard titled “Transport Layer Security (TLS) Transport Model for the Simple Network Management Protocol (SNMP)” [Har10] has been approved. For implementations running SNMP over TCP and not otherwise securing SNMP with IPsec or SSH, the methods in [Har10] are RECOMMENDED.

#### 5.4.4 Specifications for Using ISMS with SSH

- The SSH transport model does not specify cryptographic methods to be used by SSH2. Therefore, the specifications in Section 5.4.2 MUST be followed.
- The SNMP “auth none” option MUST NOT be used with the SSH transport model, and SSH MUST NOT be configured with the MAC “none” option.
- The most secure method of authentication, the public-key option, is RECOMMENDED. The public key validation step MUST NOT be skipped.
- Notification senders MUST verify the name and security credentials of notification receivers.

- SSH does not have a session resumption feature. Therefore, to minimize key agreement overhead, SSH sessions may be kept open during idle intervals.

## 5.5 Secure Web-Based Management

### 5.5.1 Components of Web Services

When Web Services become more complicated than what can be accomplished with a single server, the need exists for multiple “back end systems” to communicate with each other to provide these Web services, and new security issues surface. The security services needed for this type of architecture include authorization, access control, and single sign-on as well as identification, authentication, message integrity, and confidentiality. This section describes some of the standards-based security components and terminology associated with such Web Services. It does not cover the wide range of vulnerabilities resulting from insecurely coded Web applications that allow exploitation of unverified user inputs or software faults in common application platforms. To understand these aspects, see, for example, [SP11].

A Web Service is identified with a URI [BFM05].

Extensible Markup Language (XML) [XML08, XMLS004, XMLS104, XMLS204] is a platform-independent data format that uses HTML-like tags to describe information. It allows structured data to be shared among heterogeneous applications and systems without requiring translation.

Simple Object Access Protocol (SOAP) [SOAP007, SOAP107, SOAP207, SOAP03] is an XML- and HTTP-based protocol [Fie99, KL00, Res00] for networked procedure calls between application components. SOAP 1.2 is a product of the W3C XML Protocol (XP) Working Group. It uses many W3C and IETF specifications, particularly those for XML. SOAP is normally used with automatically generated, machine-to-machine transactions. Generally, HTTP is allowed through packet filters, and most proxies cannot filter based on SOAP content.

A SOAP message is an XML document with a single root element named “Envelope” in the SOAP envelope namespace. The envelope identifies the SOAP version, and holds the SOAP header and body. Typical SOAP prefixes are:

- SOAP Envelope env:
- SOAP encoding (SOAP 1.1) enc:
- XML Schema (datatypes) xsd:
- XML Schema instance xsi:
- WSDL wsdl:
- XML digital signatures ds:
- Web Service security wsse:

Web Services Description Language (WSDL) [WSDLp07, WSDL107, WSDL207, WSDL307] describes, in XML, how to develop a Web Services client. It includes the URI of the service, target objects, methods for those objects, parameter names and types, and return value types. WSDL uses object-oriented constructions to define services, ports, port types, operations, message values, and types. It does not have ways to define security constraints or access requirements.

UDDI (Universal Description, Discovery, and Integration) [UDDI04] is the commonly used discovery protocol for Web Services.

### 5.5.2 SDOs for Web Services

Multiple standards development organizations have developed open standards for Web Services (WS) and Web Services Security (WSS):

1. The Internet Engineering Task Force (IETF) has specified standards for HTTP 1.0/1.1 (RFC 2616), LDAPv3, TLS, and URIs.
2. The W3C (World Wide Web Consortium) has written standards for Extensible Markup Language (XML), XML-Signature Syntax and Processing (XMLDSIG), XML Encryption Syntax and Processing (XMLENC), Simple Object Access Protocol (SOAP), XML Key Management Specification (XKMS), Web Services Description Language (WSDL), Canonical XML, Exclusive XML Canonicalization, and XML Schema.
3. The Organization for the Advancement of Structured Information Standards (OASIS) has developed UDDI, Web Services Security (WS-Security), Security Association Markup Language (SAML), Web Services Reliable Messaging (WSRM), Extensible Access Control Markup Language (XACML), and a variety of PKI-oriented documents. They have also developed Web Services Distributed Management—Management of Web Services (WSDM-MOWS) and Web Services Distributed Management—Management Using Web Services (WSDM-MUWS).
4. Liberty Alliance, which has worked on federated identity and single sign-on, is a vendor consortium involved in WSS. Also, the Web Services Interoperability Organization (WS-I), ANSI, and ISO have also written standards that are potentially relevant.

### 5.5.3 Web Services Security Overview

Threats against Web Services (or SOAP) include:

- Eavesdropping on transactions by tapping into LAN or WAN connections, compromising SOAP intermediaries, or compromising servers
- Modifying requests or responses by compromising SOAP intermediaries, compromising clients, or hijacking TCP sessions
- Compromising clients' passwords
- IP address spoofing

- XML code injection
- Cross-site scripting
- Impersonating a WS server through DNS cache poisoning, DNS spoofing, or posting bogus WSDL files
- Replay of requests or responses
- Denial of service against servers or specific clients
- Traffic analysis

Three viable approaches to SOAP security exist:

1. SOAP and WS can run over IPsec VPNs. This can achieve host-to-host communications security.
2. SOAP over HTTP can use TLS, which provides communications security from SOAP application to SOAP application.
3. SOAP can use the XML Security standards for fine-grained, application-layer security, end to end.

The main drawback of the first two approaches is that they do not provide end-to-end security for portions of a message sent to multiple servers or back-end systems.

WS-Security [WSsec06] provides mechanisms implemented in SOAP to enhance SOAP messaging security with authentication, integrity, and confidentiality services.

The first component of WS-Security is XML signatures. An XML-Signature [XMLSIG08, CXML08] can be applied to any portion of an XML document. It can be based on shared secret keys and a MAC (typically a keyed hash) or on public-key-based digital signatures (e.g., RSA).

WS-Security defines security tokens that may specify identities or claims about possession of a key. They can be signed with XML-Signature.

XML-Encryption [XMLENC10] defines a mechanism to provide confidentiality for portions of an XML document. It is designed to work together with XML-Signature.

Security Assertion Markup Language (SAML) [SAML05, SAMLbp03, SAMLco05, SAMLgl05, SAMLsp05, SAMLto08] consists of XML and SOAP services, data structures, and protocols for exchanging identification, authentication, and authorization information. It is based on XML and SOAP, and it defines requests, responses, and faults. It does not use SOAP remote procedure calls. A typical use of SAML is to support access control decisions based on an identity.

XACML (XML Access Control Markup Language) [XACML05] is an OASIS standard for expressing access controls in terms of subjects, resources, and actions.

XKMS (XML Key Management Specification) [XKMS05] is a W3C specification that defines abstract interfaces to an underlying PKI. It has two parts: (1) X-KRSS (XML Key

Registration Service Specification) for public key registration and revocation and (2) X-KISS (XML Key Information Service Specification) for locating and validating keys.

Because Web Services typically use multiple servers, single sign-on, which avoids requiring users to re-authenticate to each server, is often an important requirement.

#### 5.5.4 Web Services Security Protocol Stack

The following list shows the different places where security fits into a WS protocol stack:

- Either IPsec at the Network Layer or TLS at the Transport Layer **MUST** be used with IPv4 or IPv6.
- Application Transport usually consists of running everything over HTTP, so the usual firewall settings for the Web **SHOULD** be applied.
- Many WS applications rely on the Domain Name System (DNS) to discover servers, store and retrieve certificates or schemas, and perform other operations. Therefore, secure operation of the DNS and protection of DNS servers against denial of service attacks are critical components of WS security. The deployment of DNS **SHOULD** follow the guidelines in [CR10].
- Application Descriptions written in WSDL **MUST** be secured with XML-DSIG.
- Application Messaging with SOAP over XML **SHOULD** be secured with XML-DSIG, XML-ENC, and XKMS. They **SHOULD** be secured with WS-Security, SAML (for identity management, authentication, and authorization), and XACML (for Application Layer access control) where these protocols apply. They **MAY** also be secured with WS-Security WSDL.
- Discovery with UDDI over XML **MUST** be secured with IPsec, TLS, or XML-DSIG and XML-ENC.
- For PKI, the IETF's Internet X.509 PKIX [Coo08] **MUST** be used.

#### 5.5.5 Web Services Security Profile

This section identifies WS and WSS standards. Each sub-section identifies the working group leading the development of the standard.

##### 5.5.5.1 Internet Engineering Task Force (IETF)

Standard	Type	Status	References
HTTP 1.0 and 1.1	Core	Stable	RFC 2616
LDAPv3	Support	Stable	RFC 3673
SysLog	Support	Revision	OIF draft IA [Syslog12]
IPsec	Security	Revision	OIF draft IA [Sec12]
TLS	Security	Revision	RFC 5426



URI	Core	Stable	RFC 3617
-----	------	--------	----------

**Table 2: Web Services Security Standards from the IETF.**

#### 5.5.5.2 World Wide Web Consortium (W3C)

Standard	Type	Status	References
XML	Core	Stable	[XML08], [CXML08]
SOAP	Core	Version 2.0 in progress	[SOAP007], [SOAP107], [SOAP207], [SOAPT03]
WSDL	Core	Stable	[WSDLp07], [WSDL107]
XML Schema	Core	Stable	[XMLS004], [XMLS104], [XMLS204]
XML-DSIG	Security	Stable	[XMSIG08]
XML-ENC	Security	Stable	[XMLENC10]
XKMS	Support	Stable	[XKMS05]

**Table 3: Web Services Security Standards from the W3C.**

#### 5.5.5.3 Profile of XML-DSIG

All of the stipulations in this standard [XMSIG08] apply. In addition, the following notes profile the XML-DSIG specification for use in securing Web Services based management systems:

- See RFC 2807 for XML signature requirements.
- The reference to RFC 1750 is updated to RFC 4086 [ESC05].
- This specification defines both a MAC based on HMAC-SHA-1 and a shared secret and a digital signature scheme based on a hash function and a public key system. The former is more efficient and ideally suited to protecting messages on communications

channels between two parties, but it needs a shared secret. The latter is more suitable to authenticating messages in store-and-forward systems, messages that may exist persistently and need to be authenticated at undetermined future times, and messages that may need to be authenticated by multiple parties.

- In all cases, MD5 **MUST NOT** be used.
- Applications **SHOULD** use time stamps or increasing message IDs to help identify replays.
- Implementation of Base64 [FB96], XPath node-sets [XPath07], and XML-C14N canonicalization [CXML08] is **REQUIRED**.
- URI attributes **MUST NOT** include fragment identifiers.
- The PGPData, SPKIData, and MgmtData elements **SHOULD NOT** be used.

For public-key based signatures:

- DSA signatures are **RECOMMENDED** over RSA signatures because they are shorter, are more efficient for the signer, and allow pre-computation. RSA signatures have more flexible key sizes and are more efficient for the verifier. They **MAY** be used if these characteristics are critical for the security or application requirements.
- Applications are encouraged to use the Manifest element together with multiple reference objects to reduce the number of more computationally expensive public key operations required.
- Applications **SHOULD** use keyInfo with type X509Data. The RetrievalMethod element is **RECOMMENDED** to keep messages short.

#### 5.5.5.4 *Profile of XML Encryption Syntax and Processing*

All of the stipulations in this standard [XMLENC10] apply. In addition, the following notes profile the XML-ENC specification for use in securing Web Services based management systems:

- See <http://www.w3.org/TR/xml-encryption-req> for XML encryption requirements.
- Compliance with the XML Namespace Specification [XMLNS06] is **REQUIRED**.
- Decryption **MUST** allow for up to 255 bytes of padding.
- An integrity check [XMLSIG08] **MUST** be applied to data encrypted with this standard.
- Implementation of Base64 [FB96], XPath node-sets [XPath07], and XML-C14N canonicalization [CXML08] is **REQUIRED**.
- Users are cautioned that the EncryptionMethod, KeyInfo, and EncryptionProperties elements may reveal some information about the encryption process.
- The following algorithms **MUST** be implemented:

- Block Encryption: AES-128
- Key Transport: RSA-OAEP (1536-bit or longer RSA is RECOMMENDED; 1024-bit or longer RSA is REQUIRED)
- Symmetric Key Wrap: AES-128
- Message Digest: SHA-256 (SHA-1 and SHA-512 are OPTIONAL; RIPEMD and MD5 are MUST NOT be used)

#### 5.5.5.5 *Profile of XKMS 2.0*

- See <http://www.w3.org/TR/2003/NOTE-xkms2-req-20030505> for the XML Key management requirements.
- Note that this document describes an information service and a registration service for public keys used with XML-DSIG and XML Encryption. It does not address PKI issues and trust models directly.
- Clients **MUST** be configured securely with the FQDNs or IP addresses of all servers they use and with the keys used to authenticate responses. Without other means of verification, clients **MUST NOT** rely on a DNS SRV RR to discover a server. For example, if the client uses the KISS Locate service to parse a certificate and obtain a name and key, it has to trust the server to return the correct name. The same considerations apply to the KISS validate service.
- Similarly, servers **MUST** authenticate clients of the Registration service.
- Communications between clients and servers **MUST** use message-level integrity and replay detection (secured, for example, with XML-DSIG, Secure HTTP, TLS, or IPsec). Message confidentiality is **OPTIONAL**, except for the transmission of private keys, in which case it is **REQUIRED**.
- Clients and servers **MUST** implement the two-phase request protocol, which servers **SHOULD** use when they detect a possible denial of service attack, even if signed requests are used.
- Servers **MUST** require clients of the Registration service to provide proof of possession of the private key.
- Clients **SHOULD** generate keys used for digital signatures themselves.

## 5.5.6 OASIS

Standard	Type	Status	References
UDDI	Core	Stable	[UDDI04]
Web Services Security (WS-Security)	Security	Version 1.1 stable	[WSsec06]
SAML	Security	Version 2.0 stable	[SAML05], [SAMLbp05], [SAMLco05], [SAMLsp05]
XACML	Security	Stable	[XACML05]
WSS X.509 Certificate Token Profile	Security	Stable	[WSSctp06]

**Table 4: Web Services Security Standards from OASIS.**

### 5.5.6.1 Profile of WS-Security

- This profile applies to version 1.1, [WSsec06], which incorporates approved errata.
- Note that this specification defines SOAP data structures intended to be building blocks for security protocols. It does not define the security protocols themselves, and, therefore, significant effort is still required to verify that protocols using these methods are secure. See, in particular, the Security Considerations in Section 13 of [WSsec06].
- This specification covers a wide variety of security methods, so use of the subset of these methods does not ensure interoperability.
- The URI reference to RFC 2396 is updated to RFC 3986 [BFM05].
- If the same data are to be encrypted and signed, it is usually preferable to sign the encrypted data to protect against attacks that tamper with the ciphertext.
- Inclusive canonicalization SHOULD be used, except in the case that signed information will be inserted into another XML document, in which case exclusive canonicalization SHOULD be used.

### 5.5.6.2 Profile of SAML

- Enveloped XML digital signatures MUST be implemented as the primary authentication and integrity method for SAML.
- SAML protocol messages MUST be signed by the original sender.

- SAML implementations **MUST** support RSA as an XML Digital Signature mechanism.
- SAML messages **MUST** use and verify Time values to detect replay attacks. SAML assertions **SHOULD** contain valid lifetimes.
- Because of the cost of verifying digital signatures, SAML is vulnerable to Denial of Service attacks. Therefore, the origin and integrity of SAML protocol messages **SHOULD** be protected by a lower-layer security system, e.g., TLS or IPsec.
- If HTTP Basic Authentication is used, TLS **MUST** be used as well.

#### 5.5.6.3 Profile of XACML

- XACML messages **MUST** be authenticated and protected with respect to integrity and replay detection.
- XACML messages **SHOULD** be encrypted to protect confidentiality as well.
- XACML policies **MUST** be signed by the issuer of the policy.
- A “not applicable” response from a Policy Decision Point **MUST** be treated as a “deny.”

#### 5.5.7 ANSI, NIST, and ITU-T

Standard	Type	Status	References
ANSI X9.84 (XCBF)	Security	Stable	
ANSI X9.96 (XCMS)	Security	Stable	
ANSI X9.73 (CMS)	Core	Stable	
NIST SP 800-81r1	Security Support	Draft	[CR10]
ITU-T X.509	Core	Stable	

**Table 5: Web Services Security Standards from ANSI, NIST, and ITU-T.**

## 5.6 Other Protocols Supporting Security

### 5.6.1 RADIUS

RADIUS performs authentication, authorization, and accounting. It is not designed to provide confidentiality, integrity, or key management services. If these security services

are needed along with RADIUS, users MAY deploy RADIUS over IPsec or use other comparable solutions.

A NE that implements a RADIUS client to obtain user authentication information from a RADIUS server MUST use that authentication as the sole authentication of the client. These implementations MUST support RADIUS as defined in [Rig00].

RADIUS MAY be used with PAP, CHAP, UNIX login, or other authentication mechanisms. When used with PAP, RADIUS protects the PAP ID and password with a shared secret. RADIUS specifies client-to-server authentication and does not specify a server-to-client authentication mechanism. RADIUS also does not specify a user-to-client authentication mechanism. RADIUS uses a shared secret between the client and server. It does not specify how to establish or change this shared secret. If RADIUS proxy servers are used, the secret must also be shared with any participating proxy servers.

Users of RADIUS SHOULD develop an operational continuity plan for the case in which their RADIUS server becomes unavailable. Alternatives include using a local authentication database or configuring sufficient backup RADIUS servers. The security of such solutions SHOULD be evaluated in light of the possibility that a denial of service attack on the RADIUS server may be part of a broader attack on NEs. The following three specifications are taken from [Rig00]:

- RADIUS implementations SHOULD NOT use keep alives.
- RADIUS implementations SHOULD use the officially assigned UDP port of 1812.
- RADIUS implementations SHOULD use a challenge response mechanism.

Using the challenge response mechanism, the server sends a challenge message to the client consisting of a random number, and the client encrypts the random number using the shared secret and returns it to the server. The random number SHOULD be at least 16 octets. Implementations MUST have access to a source of cryptographically strong random or pseudo-random numbers. See [ESC05], [Gut98], [KSF99], and [Koç09] for additional guidelines and recommendations on generating pseudo-random numbers.

### 5.6.2 S/MIME

S/MIME provides encryption and digital signatures for MIME objects. It is used primarily for secure email and is the only system mentioned in this IA that has a built-in, protocol-based mechanism for non-repudiation of message origin. However, use of S/MIME at this time is out of the scope of this document.

### 5.6.3 syslog

Logging provides an often indispensable tool to reconstruct events and isolate problems after they have occurred. The OIF's Implementation Agreement on Logging and Auditing with Syslog [Syslog12] describes flexible methods for configuring and then securing syslog. Secure logging SHOULD be used to record all security events, error conditions, and configuration changes.

## 6. Objectives Satisfied by Security Systems

Table 6 provides details on which objectives from Section 3 are satisfied by using the security systems as specified in Sections 5.1 through 5.5. A ‘√’ indicates that the objective is satisfied by the security system. ‘May’ indicates that satisfaction of the objective is dependent upon the vendor’s specific implementation of the security system.

**Table 6: Applicability of Security Solutions to Different Interfaces.**

<b>Objective</b>	<b>SNMPv3</b>	<b>TLS</b>	<b>SSH</b>	<b>IPsec</b>	<b>WSS</b>
C-1	√	√	√	√	√
C-2		√	√	√	√
C-3	May	May	May	May	May
C-4		May	May	√	May
I-1	√	√	√	√	√
I-2	Note 1	Note 2	√	√	√
I-3		May	May	May	May
I-4	Note 1			√ within preset window	√
K-1	Note 3	√	√	√	√
K-2	Note 3	√	√	√	√
K-3	Note 3		May	√	
K-4	Note 4	√, using resume session		√	√, key life-times
A-1	√	√	√	√	√
A-2	√	√	√	√	√
N-1	√	May	May	√	May
N-2	Note 5	√	√	√	May
R-1				Note 6	√
AC-1	√	Note 2	Note 2		√
AC-2	Note 2	Note 2	Note 2		√
L-1	May	May	May		May
L-2	May	May	May		May
L-3	May	May	May		√
L-4	May	May	May		√
L-5	√	May	May	May	
L-6	√	√		√	May
D-1	Note 7	Note 7	Note 7	Note 7	Note 7
D-2	Note 7	Note 7	Note 7	Note 7	Note 7
T-1		May	May	√	May
T-2				√	

Note 1: This objective can be satisfied by using the Timeliness Value.

Note 2: This objective can be satisfied by using TCP Wrappers at the server.

Note 3: To satisfy this objective, a secure key distribution protocol (e.g., IKEv2) needs to be implemented: IKEv2 can satisfy K-1, K-2, and K-3.

Note 4: This objective can be satisfied by using pre-placed initial keys and the rekeying option.

Note 5: N-2 may be satisfied, fully or partially, by using certain key management protocols (e.g., based on IKE) with SNMPv3.

Note 6: Support for non-repudiation of message origin can be provided by using an asymmetric (digital signature) algorithm for the integrity check (which has been proposed for multicast groups).

Note 7: In all cases of denial of service objectives D-1 and D-2, the degree to which these objectives are satisfied depends upon the implementation and configuration, not the protocol. In the design of IPsec and WSS, certain explicit choices were made to reduce the impact of denial of service attacks. SNMPv3 has the potential advantage over the others that it does not rely on the costly public key computations that can overload processing capability.

## 7. Summary

This version 2.0 brings the OIF's *Security for Management Interfaces to Network Elements* up to date with (1) new work in the OIF on control plane security and on logging and auditing with syslog; (2) new work in the IETF on IPsec, IKE, TLS, SSH, and ISMS; and (3) ongoing work in other SDOs on security for network management, in particular, for securing management interfaces based on Web Services and XML.

## 8. References

### 8.1 Normative References

The following references contain provisions that, through reference in this text, constitute provisions of this specification. At the time of publication, the editions indicated were valid. Many references are subject to revision, and parties to agreements based on this implementation agreement are encouraged to investigate the possibility of applying the most recent editions of the references indicated below.

- [Bad09] Badra, M., "Pre-Shared Key Cipher Suites for TLS with SHA-256/384 and AES Galois Counter Mode," IETF RFC 5487, March 2009.
- [BB12] Bider, D., and M. Baushke, "SHA-2 Data Integrity Verification for the Secure Shell (SSH) Transport Layer Protocol," IETF RFC 6668, July 2012.
- [BH09] Badra, M., and I. Hajjeh, "ECDHE\_PSK Cipher Suites for Transport Layer Security (TLS)," IETF RFC 5489, March 2009.



- [BFM05] Berners-Lee, T., R. Fielding, and L. Masinter, “Uniform Resource Identifier (URI): Generic Syntax,” IETF RFC 3986, January 2005.
- [BKN06] Bellare, M., T. Kohno, and C. Namprempre, “The Secure Shell (SSH) Transport Layer Encryption Modes,” IETF RFC 4344, January 2006.
- [B-W03] Blake-Wilson, S., et al., “Transport Layer Security (TLS) Extensions,” IETF RFC 3546, June 2003.
- [B-W06] Blake-Wilson, S., et al., “Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS),” IETF RFC 4492, May 2006.
- [Bra97] Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” IETF RFC 2119, March 1997.
- [Blu04] Blumenthal, B, F. Maino, and K. McCloghrie, “Advanced Encryption Standard (AES) Cipher Algorithm in the SNMP User-based Security Model,” IETF RFC 3826, June 2004.
- [BW02] Blumenthal, U., and B. Wijnen, “User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3),” IETF RFC 3414, December 2002.
- [CF06] Cusack, F., and M. Forssen, “Generic Message Exchange Authentication for the Secure Shell Protocol (SSH),” IETF RFC 4256, January 2006.
- [CHPW02] Case, J., D. Harrington, R. Presuhn, and B. Wijnen, “Message Processing and Dispatching for the Simple Network Management Protocol (SNMP),” IETF RFC 3412, December 2002.
- [CR10] Chandramouli, R., and S. Rose, *Secure Domain Name System (DNS) Deployment Guide*, NIST Special Publication SP800-81r1, April 2010.
- [Cho02] Chown, P., “Advanced Encryption Standard (AES) Ciphersuites for Transport Layer Security (TLS),” IETF RFC 3268, June 2003. This is obsoleted by TLS 1.2, RFC 5246 [DR08].
- [Coo08] Cooper, D., S. Santesson, S. Farrell, S. Boeyen, R. Housley, W. Polk, “Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile,” IETF RFC 5280, May 2008.
- [CXML08] Boyar, J., and G. Marcy, “Canonical XML Version 1.1,” W3C Recommendation, 2 May 2008, <http://www.w3.org/TR/2008/REC-xml-c14n11-20080502/>.
- [DA99] Dierks, T., and C. Allen, “The TLS Protocol,” IETF RFC 2246, January 1999. This is obsoleted by RFC 4346.
- [DR06] Dierks, T., and E. Rescorla, “The TLS Protocol, Version 1.1,” IETF RFC 4346, April 2006. This is obsoleted by RFC 5246.
- [DR08] Dierks, T., and E. Rescorla, “The TLS Protocol, Version 1.2,” IETF RFC 5246, August 2008.
- [Eas11] Eastlake, D., 3<sup>rd</sup>, “Transport Layer Security (TLS) Extensions: Extension Definitions,” IETF RFC 6066, January 2011.

- [ET05] Eronen, P., and H. Tschofenig, "Pre-Shared Key Ciphersuites for Transport Layer Security (TLS)," IETF RFC 4279, December 2005.
- [FB96] Freed, N., and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies," IETF RFC 2045, November 1996.
- [FLRW03] Frye, R., D. Levi, S. Routhier, and B. Wijnen, "Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework," IETF RFC 3584, August 2003.
- [Fie99] Fielding, R., et al., "Hypertext Transfer Protocol -- HTTP/1.1," IETF RFC 2616, June 1999.
- [FK11] Frankel, S., and S. Krishnan, "IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap," IETF RFC 6071, February 2011.
- [FPS06] Friedl, M., N. Provos, and W. Simpson, "Diffie-Hellman Group Exchange for the SSH Transport Layer Protocol," IETF RFC 4419, March 2006.
- [GL06] Galbraith, J., and P. Remaker, "The Secure Shell (SSH) Session Channel Break Extension," IETF RFC 4335, January 2006.
- [GT06] Galbraith, J., and R. Thayer, "The Secure Shell (SSH) Public Key File Format," IETF RFC 4716, November 2006.
- [GVB07] Galbraith, J., J. Van Dyke, and J. Bright, "Secure Shell Public Key Subsystem," IETF RFC 4819, March 2007.
- [HM10] Haberman, B., Ed., and D. Mills, "Network Time Protocol Version 4: Autokey Specification," IETF RFC 5906, June 2010.
- [HPW02] Harrington, D., R. Presuhn, and B. Wijnen, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks," IETF RFC 3411, December 2002.
- [HH09] Harrington, D., and W. Hardaker, "Transport Security Model for the Simple Network Management Protocol (SNMP)," IETF RFC 5591, June 2009.
- [HS09] Harrington, D., and J. Schoenwaelder, "Transport Subsystem for the Simple Network Management Protocol (SNMP)," IETF RFC 5590, June 2009.
- [HSH09] Harrington, D., J. Salowey, and W. Hardaker, "Secure Shell Transport Model for the Simple Network Management Protocol (SNMP)," IETF RFC 5592, June 2009.
- [Har06] Harris, B., "RSA Key Exchange for the Secure Shell (SSH) Transport Layer Protocol," IETF RFC 4432, March 2006.
- [Hol04] Hollenbeck, S., "Transport Layer Security Protocol Compression Methods," IETF RFC 3749, May 2004.
- [HSGW05] Hutzelman, J., J. Salowey, J. Galbraith, and V. Welch, "Generic Security Service Application Program Interface (GSS-API) Authentication and Key

- Exchange for the Secure Shell (SSH) Protocol,” IETF RFC 4462, May 2006.
- [Igo11] Igoe, K., “Suite B Cryptographic Suites for Secure Shell (SSH),” IETF RFC 6239, May 2011.
- [IS09] Igoe, K., and J. Solinas, “AES Galois Counter Mode for the Secure Shell Transport Layer Protocol,” IETF RFC 5647, August 2009.
- [IS11] Igoe, K., and D. Stebila, “X.509v3 Certificates for Secure Shell Authentication,” IETF RFC 6187, March 2011.
- [KHNE10] Kaufman, C., P. Hoffman, Y. Nir, and P. Eronen, “Internet Key Exchange Protocol Version 2 (IKEv2),” IETF RFC 5996, September 2010.
- [Ken05a] Kent, S., “IP Encapsulating Security Payload (ESP),” IETF RFC 4303, December 2005.
- [Ken05b] Kent, S., “IP Authentication Header,” IETF RFC 4302, December 2005.
- [KL00] Khare, R., and S. Lawrence, “Upgrading to TLS Within HTTP/1.1,” IETF RFC 2817, May 2000.
- [KS05] Kent, S., and K. Seo, “Security Architecture for the Internet Protocol,” IETF RFC 4301, December 2005.
- [LL06] Lehtinen, S., and C. Lonvick, “The Secure Shell (SSH) Protocol Assigned Numbers,” IETF RFC 4250, January 2006.
- [LMS02] Levi, D., P. Meyer, and B. Stewart, “Simple Network Management Protocol (SNMP) Applications,” IETF RFC 3413, December 2002.
- [MH99] Medvinsky, A., and M. Hur, “Addition of Kerberos Cipher Suites to Transport Layer Security (TLS),” IETF RFC 2712, October 1999.
- [Mil10] Mills, D., et al., “Network Time Protocol Version 4: Protocol and Algorithms Specification,” IETF RFC 5905, June 2010.
- [NN09] Narayan, K., and D. Nelson, “Remote Authentication Dial-In User Service (RADIUS) Usage for Simple Network Management Protocol (SNMP) Transport Models,” IETF RFC 5608, August 2009.
- [Res00] Rescorla, E., “HTTP over TLS,” IETF RFC 2818, May 2000.
- [Res08] Rescorla, E., “TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter Mode (GCM),” IETF RFC 5289, August 2008.
- [Rig00] Rigney, C., et al., “Remote Authentication Dial In User Service (RADIUS),” IETF RFC 2865, June 2000.
- [RRDO10] Rescorla, E., M. Ray, S. Dispensa, and N. Oskov, “Transport Layer Security (TLS) Renegotiation Indication Extension,” IETF RFC 5746, February 2010.
- [SAML05] Cantor, S., et al., “Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0,” OASIS Standard, 15 March 2005, <http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf>.

- [SAMLbp05] Cantor, S., et al., “Bindings for the OASIS Security Assertion Markup Language (SAML) V2.0,” OASIS Standard, 15 March 2005, saml-bindings-2.0-os, <http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf>.
- [SAMLco05] Mishra, P., R. Philpott, and E. Maler, “Conformance Requirements for the OASIS Security Assertion Markup Language (SAML) V2.0,” OASIS Standard, 15 March 2005, <http://docs.oasis-open.org/security/saml/v2.0/saml-conformance-2.0-os.pdf>.
- [SAMLsp05] Hirsch, F., R. Philpott, and E. Maler,, “Security and Privacy Considerations for the OASIS Security Assertion Markup Language (SAML) V2.0,” OASIS Standard, 15 March 2005, Document identifier oasis-sstc-saml-sec-consider-2.0-os, <http://docs.oasis-open.org/security/saml/v2.0/saml-sec-consider-2.0-os.pdf>
- [SCM08] J. Salowey, J., A. Choudhury, and D. McGrew, “AES Galois Counter Mode (GCM) Cipher Suites for TLS,” IETF RFC 5288, August 2008.
- [Sec1] Optical Internetworking Forum Implementation Agreement, “Security Extension for UNI and NNI version 2.1,” OIF-SEP-03.2, October 2012.
- [Syslog12] Optical Internetworking Forum Implementation Agreement, “OIF Control Plane Logging and Auditing with Syslog version 1.1,” OIF-SLG-01.2, October 2012.
- [SG06] Schlyter, J., and W. Griffin, “Using DNS to Securely Publish Secure Shell (SSH) Key Fingerprints,” IETF RFC 4255, January 2006.
- [SG09] Stebila, D., and J. Green, “Elliptic Curve Algorithm Integration in the Secure Shell Transport Layer,” IETF RFC 5656, December 2009.
- [SH11] Saint-Andre, P., and J. Hodges, “Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS),” IETF RFC 6125, March 2011.
- [TIM1] “Operations, Administration, Maintenance, and Provisioning Security Requirements for the Public Telecommunications Network: A Baseline of Security Requirements for the Management Plane,” T1.276-2003, July 2003.
- [TWMP07] Taylor, D., T. Wu, N. Mavrogiannopoulos, and T. Perrin, “Using the Secure Remote Password (SRP) Protocol for TLS Authentication,” IETF RFC 5054, November 2007.
- [UDDI04] Clement, L., et al., “UDDI Version 3.0.2,” OASIS UDDI Spec Technical Committee Draft, October 2004, <http://uddi.org/pubs/uddi-v3.0.2-20041019.htm>.
- [WPM02] Wijnen, B., R. Presuhn, and K. McCloghrie, “View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP),” IETF RFC 3415, December 2002.

- [WSsec06] Nadalin, A., et al., eds., “Web Services Security: SOAP Message Security 1.1 (WS-Security 2004),” OASIS Standard incorporating Approved Errata, November 2006, <http://docs.oasis-open.org/wss/v1.1/wss-v1.1-spec-errata-os-SOAPMessageSecurity.pdf>.
- [WSDLp07] Booth, D., and C. Liu, “Web Services Description Language (WSDL) Version 2.0 Part 0: Primer,” W3C Recommendation 26, June 2007, <http://www.w3.org/TR/2007/REC-wsdl20-primer-20070626/>.
- [WSDL107] Chinnici, R., et al., “Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language,” W3C Recommendation 26, June 2007, <http://www.w3.org/TR/2007/REC-wsdl20-20070626>.
- [WSDL207] Chinnici, R., et al., “Web Services Description Language (WSDL) Version 2.0 Part 2: Adjuncts,” W3C Recommendation 26, June 2007, <http://www.w3.org/TR/2007/REC-wsdl20-adjuncts-20070626/>.
- [WSDL307] Vedamuthu, A., “Web Services Description Language (WSDL) Version 2.0 SOAP 1.1 Binding,” W3C Working Group Note 26, June 2007, <http://www.w3.org/TR/2007/NOTE-wsdl20-soap11-binding-20070626>.
- [WSSctp06] Nadalin, A., et al., eds., “Web Services Security X.509 Certificate Token Profile 1.1,” OASIS Standard Specification, 1 February 2006, <http://docs.oasis-open.org/wss/v1.1/wss-v1.1-spec-os-x509TokenProfile.pdf>.
- [XACML05] Moses, T., “eXtensible Access Control Markup Language (XACML) Version 2.0,” OASIS Standard oasis-access\_control-xacml-2.0-core-spec-os, 1 February 2005, [http://docs.oasis-open.org/xacml/2.0/access\\_control-xacml-2.0-core-spec-os.pdf](http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf).
- [XKMS05] Hallam-Baker, P., “XML Key Management Specification (XKMS 2.0) Version 2.0,” W3C Recommendation, 28 June 2005, <http://www.w3.org/TR/2005/REC-xkms2-20050628/>.
- [XML08] Bray, T., “Extensible Markup Language (XML) 1.0 (Fifth Edition), W3C Recommendation 26, November 2008, <http://www.w3.org/TR/2008/REC-xml-20081126/>.
- [XMLENC11] Imamura, T., et al., “XML Encryption Syntax and Processing Version 1.1,” W3C Editor’s Draft 03, July 2012. <http://www.w3.org/2008/xmlsec/Drafts/xmlenc-core-11/>.
- [XMLNS06] Bray, T., D. Hollander, and A. Layman, “Namespaces in XML 1.1 (Second Edition),” W3C Recommendation, 16 August 2006, <http://www.w3.org/TR/2006/REC-xml-names11-20060816>.
- [XMLSIG08] Bartel, M., et al., “XML Signature Syntax and Processing (Second Edition),” W3C Recommendation, 10 June 2008, <http://www.w3.org/TR/2008/REC-xmldsig-core-20080610/>.

- [XPath10] Berglund, A., et al., “XML Path Language (XPath) Version 2.0 (Second Edition),” W3C Recommendation, 14 December 2010, <http://www.w3.org/TR/xpath20/>. [Link errors corrected 3 January 2011.]
- [YL06a] Ylönen, T., and C. Lonvick, “The Secure Shell (SSH) Protocol Architecture,” IETF RFC 4251, January 2006.
- [YL06b] Ylönen, T., and C. Lonvick, “The Secure Shell (SSH) Transport Layer Protocol,” IETF RFC 4253, January 2006.
- [YL06c] Ylönen, T., and C. Lonvick, “The Secure Shell (SSH) Authentication Protocol,” IETF RFC 4252, January 2006.
- [YL06d] Ylönen, T., and C. Lonvick, “The Secure Shell (SSH) Connection Protocol,” IETF RFC 4254, January 2006.

## 8.2 Informative References

- [ANSI95] “Synchronous Optical Network (SONET) Data Communications Channel Protocols and Architectures,” ANSI T1-105-04, 1995.
- [ATMF02] Methods for Securely Managing ATM Network Elements—Implementation Agreement, The ATM Forum, AF-SEC-0179.000, April 2002.
- [BSB05] Barrett, D.J., R. E. Silverman, and R. Byrnes, *SSH, The Secure Shell: The Definitive Guide, Second Edition*, O'Reilly Media, May 2005.
- [CC] See <http://www.commoncriteriaportal.org/thecc.html>.
- [DH98] Deering, S., and R. Hinden, “Internet Protocol, Version 6 (IPv6) Specification,” IETF RFC 2460, December, 1998.
- [ESC05] Eastlake, D., J. Schiller, and S. Crocker, “Randomness Requirements for Security,” IETF RFC 4086, June 2005.
- [FCK96] Freier, A.O., P. Carlton, and P.C. Kocher, “The SSL Protocol Version 3.0,” <http://tools.ietf.org/html/draft-ietf-tls-ssl-version3-00>, November 1996.
- [Gut98] Gutmann, P., “Software Generation of Practically Strong Random Numbers,” *Seventh USENIX Security Symposium Proceedings*, The USENIX Association, 1998, pp. 243–257.
- [Har10] Hardaker, W., “Transport Layer Security (TLS) Transport Model for the Simple Network Management Protocol (SNMP),” IETF RFC 5963, August 2010.
- [ISO] See, in particular, the three parts of ISO/IEC 15408 available from <http://standards.iso.org/ittf/PubliclyAvailableStandards/>. These are being replaced by the ISO 27000 series.
- [ITUDCN] Architecture and Specification of Data Communication Network, ITU-T Rec. G.7712/Y.1703, Edition 4.0, September 2010.
- [JLN11] Jankiewicz, E., J. Loughney, and T. Narten, “IPv6 Node Requirements,” IETF RFC 6434, December 2011.

- [KSF99] Kelsey, J., B. Schneier, and N. Ferguson, “Notes on the Design and Analysis of the Yarrow Cryptographic Pseudorandom Number Generator,” *Sixth Annual Workshop on Selected Areas in Cryptography*, Springer-Verlag, 1999.
- [Koç09] Koç, Ç., ed., *Cryptographic Engineering*, Springer, 2009.
- [NISTmodes] “Recommendation for Block Cipher Modes of Operation,” NIST Special Publication 800-38A, U.S. Government Printing Office, Washington, DC, July 2001, updated with 800-38C in May 2004, 800-38B in May 2005, 800-38D in November 2007, and an addendum in October 2010. For a current list of these standards, see [http://csrc.nist.gov/groups/ST/toolkit/BCM/current\\_modes.html](http://csrc.nist.gov/groups/ST/toolkit/BCM/current_modes.html).
- [OIF03] Optical Internetworking Forum Implementation Agreement, “Security for Management Interfaces to Network Elements,” OIF-SMI-01.0, September 4, 2003.
- [OIF06] Optical Internetworking Forum Implementation Agreement, “Addendum to the Security for Management Interfaces to Network Elements,” OIF-SMI-02.1, March 21, 2006.
- [OMG02] “Security Services Specification, v1.8,” Object Management Group (OMG), March 2002. <http://www.omg.org/cgi-bin/doc?formal/02-03-11.pdf>.
- [Res01] Rescorla, E., *SSL and TLS: Designing and Building Secure Systems*, Addison-Wesley, 2001.
- [SAMLgl05] Hodges, J., R. Philpott, and E. Maler, “Glossary for the OASIS Security Assertion Markup Language (SAML) V2.0,” OASIS Standard, 15 March 2005, <http://docs.oasis-open.org/security/saml/v2.0/saml-glossary-2.0-os.pdf>.
- [SAMLto08] Raguzis, N., et al., “Security Assertion Markup Language (SAML) Technical Overview V2.0,” Committee Draft 2, 25 March 2008, <http://www.oasis-open.org/committees/download.php/27819/sstc-saml-tech-overview-2.0-cd-02.pdf>.
- [Shi07] Shirey, R., “Internet Security Glossary, Version 2,” IETF RFC 4949, August 2007.
- [SOAP007] Mitra, N., and Y. Lafon, “SOAP Version 1.2 Part 0: Primer (Second Edition),” W3C Recommendation 27, April 2007, <http://www.w3.org/TR/2007/REC-soap12-part0-20070427/>.
- [SOAP107] Gudgin, M., “SOAP Version 1.2 Part 1: Messaging Framework (Second Edition),” W3C Recommendation 27, April 2007, <http://www.w3.org/TR/2007/REC-soap12-part1-20070427/>.
- [SOAP207] Gudgin, M., “SOAP Version 1.2 Part 2: Adjuncts (Second Edition),” W3C Recommendation 27, April 2007, <http://www.w3.org/TR/2007/REC-soap12-part2-20070427/>.

- [SOAPT03] Haas, H., “SOAP Version 1.2 Specification Assertions and Test Collection (Second Edition),” W3C Recommendation 27, April 2007, <http://www.w3.org/TR/2007/REC-soap12-testcollection-20070427/>.
- [SP11] Stuttard, D., and M. Pinto, *The Web Application Hacker’s Handbook, Second Edition*, Wiley Publishing, Inc., 2011
- [UNI2.0] Optical Internetworking Forum Implementation Agreement, “User Network Interface (UNI) 2.0 Signaling Specification, OIF-UNI-02.0-Common—User Network Interface (UNI) 2.0 Signaling Specification: Common Part,” OIF-UNI-02.0-Common, February 25, 2008.
- [XMLS004] Fallside, D., and P. Walmsley, “XML Schema Part 0: Primer Second Edition,” W3C Recommendation 28, October 2004, <http://www.w3.org/TR/2004/REC-xmlschema-0-20041028/>.
- [XMLS104] Thompson, H., et al., “XML Schema Part 1: Structures Second Edition,” W3C Recommendation 28, October 2004, <http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/>.
- [XMLS204] Biron, P., and A. Malhotra, “XML Schema Part 2: Datatypes Second Edition,” W3C Recommendation 28, October 2004, <http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/>.
- [YI96] Ylönen, T., “SSH—Secure Login Connections over the Internet,” *Proceedings of the Sixth USENIX Security Symposium*, July 1996, pp. 37–42.



## Appendix A: Glossary

A thorough glossary of Internet and TCP/IP security terminology can be found in [Shi07].

## Appendix B: OIF Members When the Document Was Approved

Acacia Communications	ADVA Optical Networking
Agilent Technologies	Alcatel-Lucent
Altera	AMCC
Amphenol Corp.	Anritsu
Applied Communication Sciences	AT&T
Avago Technologies Inc.	Broadcom
Brocade	Centellax, Inc.
China Telecom	Ciena Corporation
Cisco Systems	ClariPhy Communications
Cogo Optronics	Comcast
Cortina Systems	CyOptics
Dell, Inc.	Department of Defense
Deutsche Telekom	Emcore
Ericsson	ETRI
EXFO	FCI USA LLC
Fiberhome Technologies Group	Finisar Corporation
France Telecom Group/Orange	Fujitsu
Fundacao.CPqD	Furukawa Electric Japan
GigOptix Inc.	Hewlett Packard
Hitachi	Hittite Microwave Corp
Huawei Technologies	IBM Corporation
Infinera	Inphi
IPtronics	JDSU
Juniper Networks	Kandou
KDDI R&D Laboratories	Kotura, Inc.
LeCroy	LSI Corporation
Luxtera	M/A-COM Technology Solutions, Inc.
Marben Products	Metaswitch
Mindspeed	Mitsubishi Electric Corporation
Molex	MoSys, Inc.
NEC	NeoPhotonics
Nokia Siemens Networks	NTT Corporation
Oclaro	Optoplex
PETRA	Picometrix
PMC Sierra	QLogic Corporation
Reflex Photonics	Semtech
SHF Communication Technologies	Sumitomo Electric Industries
Sumitomo Osaka Cement	TE Connectivity
Tektronix	Tellabs
TeraXion	Time Warner Cable

TriQuint Semiconductor  
Verizon  
Xilinx  
Yamaichi Electronics Ltd.

u2t Photonics AG  
Vitesse Semiconductor  
Xtera Communications