**Contribution Number: oif2002.375.06**

**Working Group: OAM&P**

**TITLE: Security for Management Interfaces to Transport Network Elements**

DATE: April 21, 2003

**SOURCE: Renée Esposito, Booz Allen Hamilton, esposito_renee@bah.com**
**Richard Graveman, Department of Defense, rfg@acm.org**
**Brian Hazzard, PhotonEx, bhazzard@phontonex.com**

**Document Status:** Straw ballot IA

**Project Name:** Security for Management Interfaces to Optical Network Elements and Auditing & Logging for Optical Network Elements

**Project Number:** oif2002.346.04

**ABSTRACT:** This IA lists objectives for securing OAM&P interfaces to a Transport Network Element and then specifies ways of using security systems (e.g., IPsec or TLS) for securing these interfaces. It summarizes how well each of the systems, used as specified, satisfies the objectives.



For additional information contact:
The Optical Internetworking Forum, 39355 California Street,
Suite 307, Fremont, CA 94538
510-608-5990 phone, info@oiforum.com

**List of Contributors**

Gary Buda, Booz Allen Hamilton
buda_gary@bah.com

Renée Esposito, Booz Allen Hamilton
esposito_renee@bah.com

Richard Graveman, Telcordia and Department of Defense
rfg@acm.org

Brian Hazzard, PhotonEx
bhazzard@photonex.com

Scott McNown, Department of Defense
srmcnow@alpha.ncsc.mil

John Naegle, Department of Defense
jhnaegl@alpha.ncsc.mil

Dimitrios Pendarakis, Tellium
DPendarakis@tellium.com

Tom Tarman, Sandia National Labs
tdtarma@sandia.gov

*TABLE OF CONTENTS*

## *LIST OF TABLES*



## *LIST OF FIGURES*

# Security for Management Interfaces to Transport Network Elements

# 1 Document Summary

This Implementation Agreement (IA) consists of two main parts. The first (Section 4) lists objectives for securing the protocols used over OAM&P interfaces to an Optical Network Element (TNE). The second (Sections 5–7) presents a model for securing these protocols at different layers, describes systems that are well-suited to secure these interfaces at various protocol layers, gives specifications for using these security systems appropriately, and summarizes how such security systems achieve the objectives in the first part. 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 specifying how to use these security systems simply and effectively to achieve as many of the listed security objectives as possible.

## 1.1 Working Group

OAM&P Working Group.

## 1.2 Problem Statement

The OIF has addressed security in its UNI and NNI specifications, which describe how ONEs use various control protocols for signaling, routing, and discovery. ONEs, however, typically have one or more (in some cases many) OAM&P interfaces used for network management, billing and accounting, configuration, maintenance, and other administrative activities. Remote access to the ONE through these OAM&P interfaces is frequently a requirement. Securing the control protocols while leaving these OAM&P interfaces unprotected opens up a huge security vulnerability. 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, packet sniffers picking up passwords, active hijacking attacks on TCP connections, and a variety of denial of service attacks. Therefore, in addition to authenticating the human user (see [T1M1]), more sophisticated protocol security is needed for these interfaces.

## 1.3 Scope

The scope of this IA is to define security objectives for OAM&P access to ONEs and to specify how to use different 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 ONE. 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 an operation, and many ONEs and OAM&P systems do not distinguish different “user-IDs” or applications. However, in addition to the using the protocol security

methods in this IA, additional methods may be used to enforce access controls based on such distinctions.

System security of the ONEs, Network Management Systems (NMS), and Element Management Systems (EMS) are 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 addressed elsewhere. For more on information assurance requirements, system security requirements, and security-related functional requirements to which products can be developed please refer to [Tel1], [Tel2], and [IATF].

Most, perhaps all, of the material in this IA is not particular to *optical* network elements but is applicable to *any* network element (TNE) and its Management Systems. This interface is 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). (Figure 1 is taken from [T1M1].) In fact, many ONEs switch or route traffic over other types of networks besides optical. These ONEs are usually managed by a single set of OAM&P protocols running over a set of OAM&P interfaces.

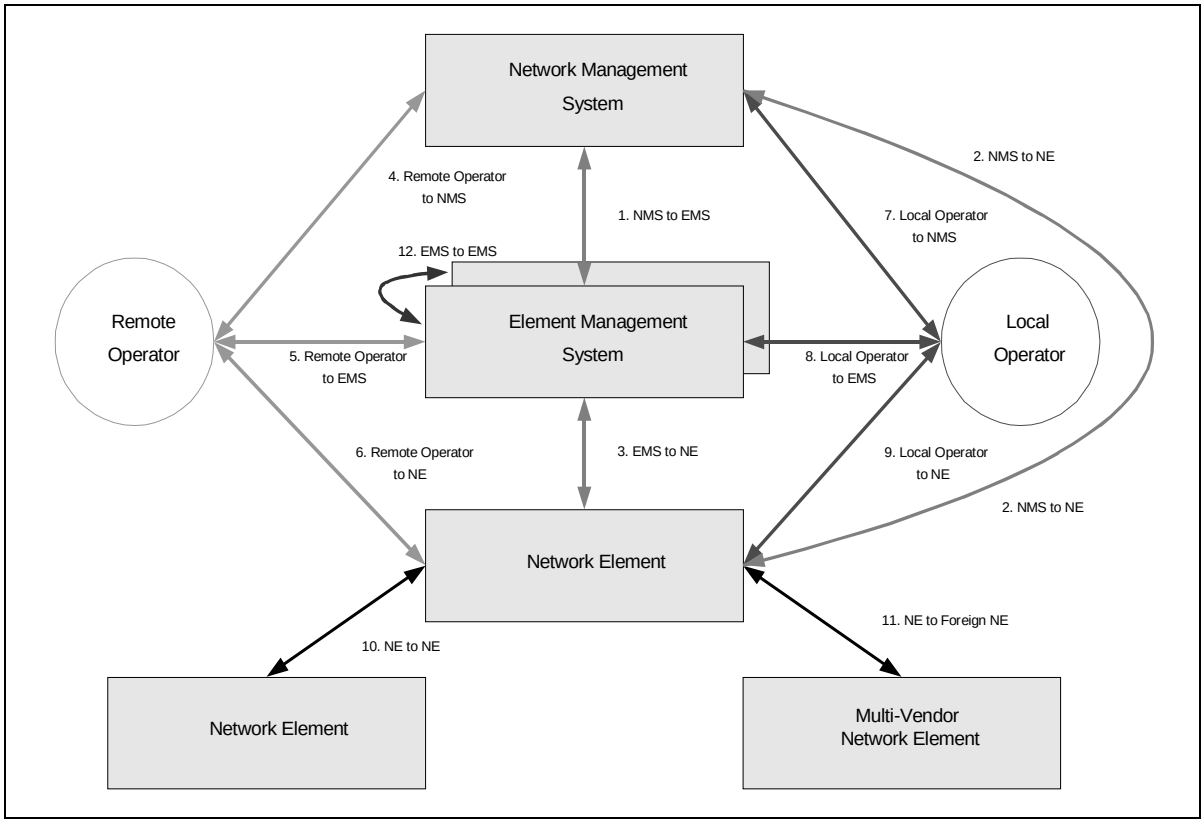


Figure 1. Network Management Security Reference Model (Nortel, T1M1).

## 1.4 Expected Outcome

The goal of this work is to complement the efforts to secure the control protocols in the UNI and NNI with a corresponding level of security for OAM&P protocols to achieve appropriate levels of interoperable security in both the control and management planes.

## 1.5 Value to OIF

This IA helps vendors (of both ONEs and Management Systems), service providers, and enterprises achieve a uniform level of security across all of the protocols used for OAM&P access to ONEs. Achieving a uniform level of security is important, because experience has shown that attacks are usually directed at the weakest point.

## 1.6 Relationship to Other Standards Bodies

- The ATM Forum has 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.
- This IA uses the RFCs written by the IETF as normative references to almost all of the security systems described herein: Kerberos, SSH, TLS, SNMPv3, IPsec, Radius, and S/MIME. The only exception is SSL.
- T1M1 has written security requirements for the management plane [T1M1]. That 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.

## 1.7 Viewpoint

This document defines no new protocols. It consolidates, profiles, and applies many aspects of work done at other standards bodies, particularly the IETF.

## 1.8 Acknowledgement

Much of the text in Sections 3–7 of this document was adapted from the ATM Forum's *Methods for Securely Managing ATM Network Elements— Implementation Agreement* [ATMF02]. The OIF acknowledges this helpful starting point and offers this document to the ATM Forum, reciprocally, for its reference and use.

# 2 Introduction

The goal of this Implementation Agreement (IA) is to apply security to management interfaces to Optical Network Elements (ONEs) 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.

## 2.1 Outline of the Implementation Agreement

This IA begins by enumerating objectives, listed in Section 4, for securing management interfaces to an ONE. The three sections after that focus on how to apply existing security systems (e.g., Kerberos, TLS, SSH, SNMPv3, or IPsec) to provide secure management access to an ONE. Section 5 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 6 briefly describes the different security systems, provides references to them, and states specifications for using them appropriately. Section 7 shows the extent to which the proper use of these security systems satisfies the objectives in Section 4. This IA does not define any new protocols or management information.

## 2.2 How to Use this Implementation Agreement

Vendors of ONEs 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 4 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.

## 2.3 Document Organization

This document is organized as follows:

- Section 3 defines the terminology and acronyms used.
- Section 4 lists and describes the security objectives.
- Section 5 describes the typical protocol stacks used by management interfaces and where security systems fit into these stacks. Among these, it recommends one choice.
- Section 6.1 covers securing protocols that run over IP with IPsec.
- Section 6.2 covers securing protocols that run over TCP with SSL or TLS.
- Section 6.3 covers securing MIB-based management systems with SNMPv3.
- Section 6.4 covers securing command line protocols with SSH.
- Section 6.5 covers securing application layer protocols with Kerberos.
- Section 6.6 covers use of these solutions together with Radius or S/MIME.
- Section 7 maps the security systems in Section 6 to the objectives in Section 4.
- Section 8 contains normative and informative references.

Each subsection of Section 6 presents a general description and specifications for using one security system aimed at satisfying the security objectives in Section 4. The table in Section 7 summarizes which security objectives from Section 4 are fulfilled by following the specifications in Section 6. The specifications for using each of the security systems in Section 6 are aimed:

1. To promote conformance of systems secured by the given security system with the security objectives in Section 4,

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.

## 2.4 Keywords

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

# 3 Terminology and Acronyms

## 3.1 Terminology

In this implementation agreement, the following definitions apply:

**Optical Network Element (ONE):** Any device supporting one or more of the defined UNI or NNI interfaces or services. It may also support other interfaces or services.

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

**Network Management System (NMS):** A terminal, network element, or system that provides services to manage an Optical 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.)

## 3.2 Acronyms

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

**AES** Advanced Encryption Standard
**CA** Certification Authority
**CBC** Cipher Block Chaining
**CMIP** Common Management Information Protocol
**CORBA** Common Object Request Broker Architecture
**CRC** Cyclic Redundancy Check
**DES** Data Encryption Standard
**DH** Diffie-Hellman
**DSS** Digital Signature Standard
**EMS** Element Management System
**ESP** Encapsulating Security Payload
**ICMP** Internet Control Message Protocol

| | |
|---|---|
| **IKE** | Internet Key Exchange |
| **IP** | Internet Protocol |
| **IPsec** | IP Security |
| **KDC** | Key Distribution Center |
| **MAC** | Message Authentication Code |
| **MIB** | Management Information Base |
| **NA** | Network Administrator |
| **NMS** | Network Management System |
| **ONE** | Optical Network Element |
| **RSA** | Rivest, Shamir, and Adleman |
| **RFC** | Request for Comments |
| **SA** | Security Association |
| **SAD** | Security Association Database |
| **SHA** | Secure Hash Algorithm |
| **S/MIME** | Secure Multipurpose Internet Mail Extensions |
| **SNMP** | Simple Network Management Protocol |
| **SPD** | Security Policy Database |
| **SSH** | Secure Shell |
| **SSL** | Secure Sockets Layer |
| **TCP** | Transmission Control Protocol |
| **TGT** | Ticket Granting Ticket |
| **TL1** | Transaction Language 1 |
| **TLS** | Transport Layer Security |
| **UDP** | User Datagram Protocol |

# 4 Threats and Security Objectives

The general threat model is that an adversary can read or write arbitrary information on the same network as the legitimate parties. In fact, these 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 by the adversary. The adversary also has full knowledge of the legitimate protocols, including security protocols, being used. However, we assume that the adversary cannot 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 often 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.
- **Unauthorized access.** Attacks under this heading include exploiting system vulnerabilities to gain access and control of system resources, compromise a network node, cause incorrect operations, modify configuration data or software, or 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 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 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 burden on the network nodes and end systems, which results 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 ONEs and Management Systems. For the purposes of this document, "interfaces between ONEs and Management Systems" is interpreted broadly to include all OAM&P communications with the ONE, regardless of the Management System endpoint. Vendors should consider the following list of security objectives and state which are met by their products.

## 4.1 Confidentiality

Confidentiality is used to protect data against partial or complete disclosure to unauthorized parties. Information that needs to be protected for confidentiality includes, but is not limited to, statistical data, configuration information, connectivity information, and management data transferred between an ONE and a Management System. Cryptography or an interface to a cryptographic device will aid in retaining information confidentiality. It must be noted that data confidentiality relies upon entity authentication and data integrity. The following are the objectives for ensuring confidentiality:

C-1 The interface between the ONE and the Management System shall support confidentiality of communications between the ONE and the Management System.

C-2 The interface between the ONE and the Management System shall support the confidentiality of passwords and keying material.

C-3 The interface between the ONE and the Management System shall support confidentiality of audit information.

C-4 The interface between the ONE and the Management System shall provide confidentiality of identities and addressing information.

## 4.2 Data Integrity

Data integrity is the ability to ensure that data have not been altered or destroyed in an unauthorized manner. For example, SNMP messages must be protected from being maliciously altered 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 data 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. The following are the objectives for ensuring data integrity:

I-1 The interface between the ONE and the Management System shall support message integrity for communications between the ONE and the Management System.

I-2 The interface between the ONE and the Management System shall support a mechanism for replay protection for communications between the ONE and the Management System.

I-3 The interface between the ONE and the Management System shall support integrity of audit information.

I-4 The interface between the ONE and the Management System shall support a mechanism to detect delay of communications between the ONE and the Management System and prohibit communications that exceed the limits of a time window.

## 4.3 Key Management

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

K-1 The interface between the ONE and the Management System shall support a 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 ONE.

K-2 The interface between the ONE and the Management System shall support a key management system for the automated and secure establishment and distribution of traffic protection keys that are shared between the Management System and the ONE.

K-3 The interface between the ONE and the Management System shall provide forward secrecy for all confidential communications between the ONE and the Management System. Forward secrecy means that compromise of long-term keys does not also compromise the contents of previous sessions that were set up using these long-term keys.

K-4 The interface between the ONE and the Management System shall provide for rekeying of traffic protection keys based on authenticated and shared key encryption keys.

## 4.4 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 is to be able to verify the identity of the user who claims to have generated a SNMP message. The following are the objectives for authentication:

A-1 The interface between the ONE and the Management System shall support the capability for each entity to establish and verify the claimed identity of the other.

A-2 The interface between the ONE and the Management System shall authenticate all communications between the ONE and the Management System.

## 4.5 Negotiation and Policy Enforcement

When each OAM&P interface is originally configured, the security policy for using this interface needs to be specified. In general, ONEs and Management Systems should be delivered from the vendor with security options enabled and appropriate warnings about disabling these options. During the establishment of a communication session, the parameters for the session must be determined. The following are the objectives for negotiation and policy enforcement:

N-1 The ONE can be configured to specify what security systems and options it requires for each OAM&P interface.

N-2 The interface between the ONE and the Management System shall provide for secure negotiation of the security services, mechanisms, and algorithms used to protect OAM&P protocols.

## 4.6 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. The following objective ensures non-repudiation:

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

## 4.7 Access Control

Access control defines and restricts the privilege to access information or perform specific functions to certain entities, roles, or systems. The following are the objectives for access control:

AC-1 The ONE shall support the capability to limit the actions of a Network Administrator (NA) based upon the NA's identity and role.

AC-2 The ONE shall support the capability to limit a NA's privileges based on the method of access.

## 4.8 Audit and Event Logging

Auditing and logging network events provides a chronological record of system activities and allows the examination of sequences of events or changes in state. The information audited and captured in a audit log may be configurable. The following are the security objectives for auditing and logging network events:

L-1 The ONE shall be capable of recording a set of events that is specified by a NA.

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

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

L-4 The ONE shall be capable of recording the identity of the NA who performed each action.

L-5 The ONE shall be 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 ONE shall be capable of detecting and reporting the occurrence of patterns, including replayed packets.

## 4.9 Denial of Service

Denial of Service (DoS) attacks can be indistinguishable from the type of network failures that a network management protocol must handle. Security protocols should be designed, insofar as possible, so as not to facilitate or enable new DoS attacks.

## 4.10 Traffic Analysis

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

T-1 The interface between the ONE and the Management System shall protect the confidentiality of parties' identities.

T-2 The interface between the ONE and the Management System shall support mechanisms that prevent an eavesdropper from learning network size, topology, or activity from an analysis of message types, lengths, counts, and timing.

# 5 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 with a SNMP agent,
- 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 management access options. For any of these management access protocols, there exist appropriate security systems described in Section 6 to provide sufficient protocol security for protecting such access from a wide range of passive and active attacks.

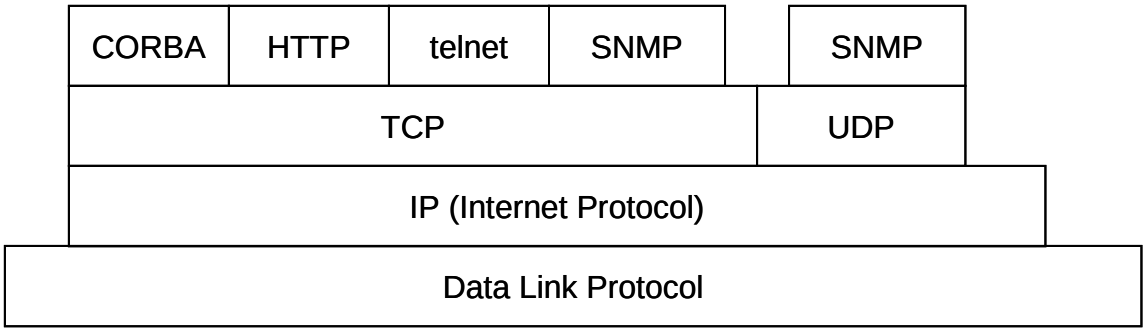


Figure 2: Typical Protocol Stacks for Management Interfaces.

## 5.1 Protocol Stacks and Security

Figure 3 is an expanded version of Figure 2 that shows where the security systems of Section 6 (shown in the shaded blocks) can fit into the protocol stack. Command line interfaces, for example, may be secured with Kerberos (within the application) or SSH (directly below the application). 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. Also, note that a choice of application layer, transport layer, or network layer security exists. For example, telnet running over TCP running over IP can be secured at any of four different layers. The unshaded blocks indicate that the security mechanisms for a given protocol are not appropriate solutions and are outside the scope of this document. For example, CORBASec [OMG02] is not a stand-alone security solution, because CORBA security rely on underlying security mechanisms, e.g., SSL.

The intent of this IA is to offer a choice of acceptable security systems and to specify how to use each appropriately to achieve security between a Management System and an ONE. To promote interoperability, one such choice is recommended.

## 5.2 Protocol Stacks and VPNs

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, there may exist an emulated link layer, but this is immaterial to the security discussion here.) VPNs operate

between routers, firewalls, or security gateways, and do not provide end-to-end security, so end-to-end security may be applied in the upper layers of Figure 4 as well.

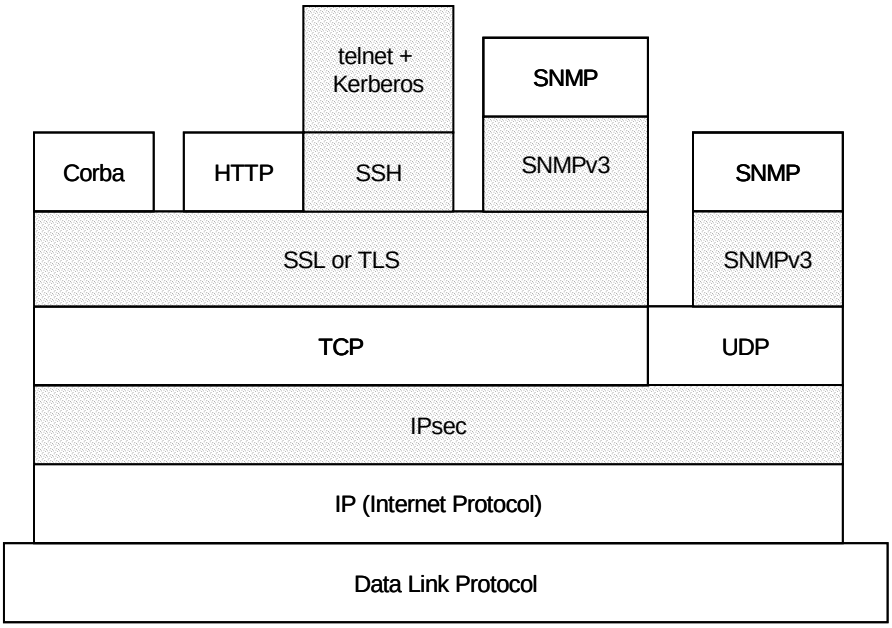


Figure 3: Protocol Stacks Including Security.

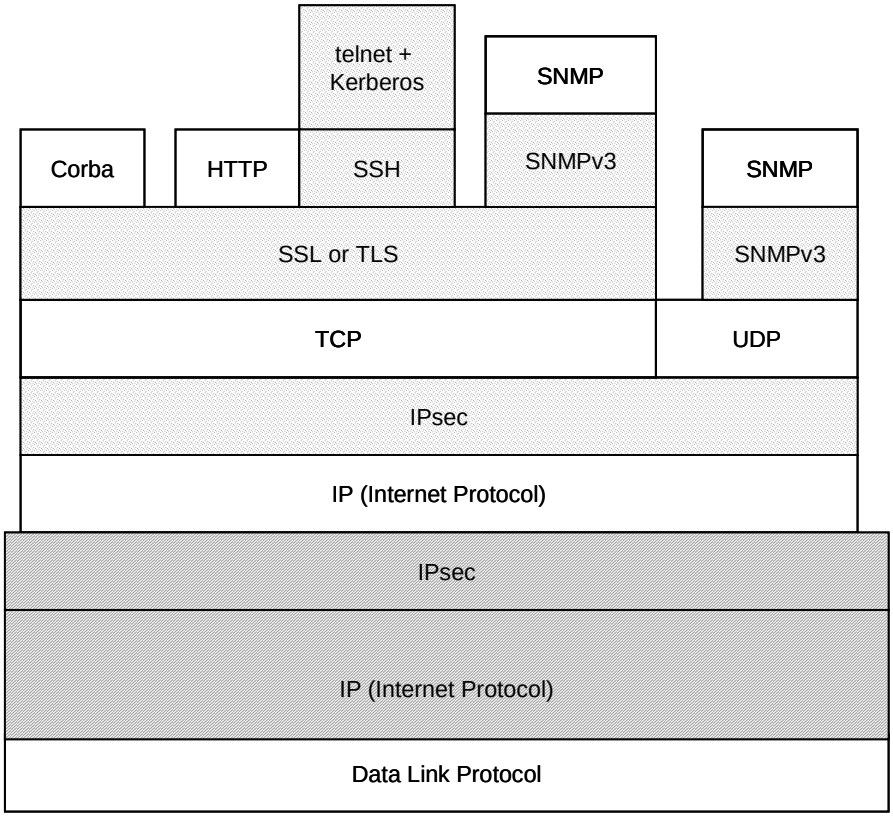


Figure 4: Protocols Stacks Including a Layer 3 VPN.

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

Security systems usually provide more than one service (e.g., authentication and integrity) and, as depicted above, they may be combined with other security protocols (see Section 6.6) to provide greater levels of security or alternative methods of authentication.

## 5.3 Management Interfaces and Security Protocols

The security systems shown in Figures 3 and 4 and Table 1 can be applied to different protocols, protocol layers, and types of OAM&P interfaces. The desired scope and security services needed may influence which security systems are chosen. For example, SSH and SSL/TLS normally protect traffic from the Management System to the ONE, 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 Figures 3 and 4, IPsec can be applied to any management interface running over IP. Therefore, the following recommendations are made:

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

- Kerberos,
- SSH,
- Lower layer protection with SSL, TLS, or IPsec.

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

Security is a requirement for *all* command line interfaces to the ONE, 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 an ONE provides MIB-based management access, it MUST support at least one of the following:

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

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

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

- SSL or TLS,
- IPsec.

IPsec is the RECOMMENDED choice and the others are OPTIONAL.

To summarize, Table 1 shows the variety of choices for protecting various management interfaces. A '√' indicates that the specified 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* ONEs [Gra03], IPsec is the RECOMMENDED solution. This choice is consistent with the fact that IPsec is mandatory in IPv6.

| Interface | Kerberos | SNMPv3 | SSL/TLS | SSH | IPSec |
|---|---|---|---|---|---|
| Web or CORBA | | | √ | | √ |
| MIB based over TCP | | √ | √ | | √ |
| MIB based over UDP | | √ | | | √ |
| Command Line | √ | | √ | √ | √ |

Table 1: Applicability of Security Solutions to Different Interfaces.

# 6 Security Systems and Specifications

## 6.1 IPsec

### 6.1.1 IPsec Description

The architecture of IPsec is defined in [KA98a]. IPsec provides cryptographic security for protocols running over IPv4 or IPv6 with the ESP (Encapsulating Security Payload, [KA98c]), the AH (Authentication Header, [KA98b]), and cryptographic key management protocols (IKE, [Pip98, MSST98, HC98]), which provide different security services. The AH transform protects IP datagrams by providing integrity 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.

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, protocol, destination address>[1], where "SPI" stands for Security Parameters Index and "protocol" is either ESP or AH. 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[2]. These parameters include the encapsulation mode (tunnel or transport), algorithms and modes of operation [NIST01], session keys, SPI value, and SA lifetime. IKE is a two-

[1] In the currently drafted IETF revisions (February 2003), only the SPI and destination address are used to identify a SA.

[2] IPsec has a mandatory provision for manual key distribution, but because manual key distribution does not allow for important functions like automatic rekeying, it is not recommended in this IA.

phase entity authentication and key management protocol[3]. It supports AH and ESP by establishing and managing the SAs in the SAD. IKE Phase 1 sets up an internal SA used to protect IKE Phase 2 cryptographic key exchanges. IKE Phase 2 creates IPsec SAs and their associated parameters and keys. IKE 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 or tunnel mode. When IPsec is used with IPv4, the protocol field in the IPv4 header contains the value for "ESP" or "AH." In Transport 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 Figures 5a and 5b.

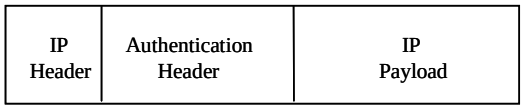


**Figure 5a: AH in Transport Mode**

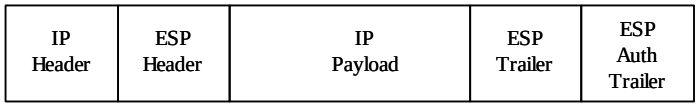


**Figure 5b: ESP in Transport Mode**

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 Figures 6a and 6b. Transport mode imposes less overhead, but it can only be used end to end and cannot be applied at routers or firewalls.

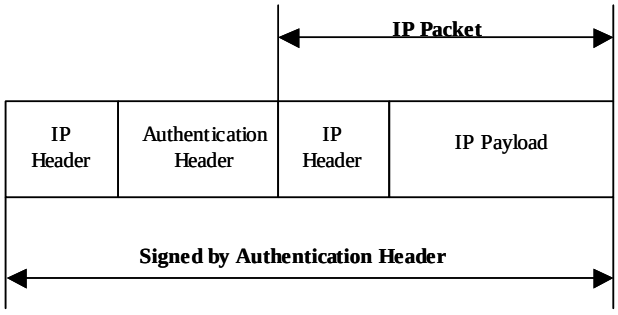


**Figure 6a: AH in Tunnel Mode**

[3] An Internet Draft for a revised and simplified version of IKE, called IKEv2, exists.

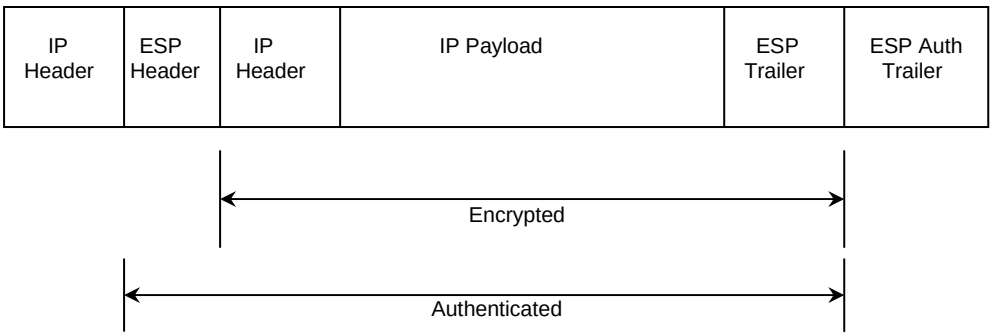


Figure 6b: ESP in Tunnel Mode.

Authentication of the parties using IPsec is implied by the possession of the integrity and confidentiality keys used by AH or ESP. Therefore, 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 to the key management, and authentication is achieved by examining the issuer, subject name, and other pertinent information in such certificates, chains of certificates, associated revocation lists, etc. Alternatively, authentication may follow from the use of IKE with pre-shared keys. Pre-shared keys require that the key value be administratively configured into each such pair of peers in a secure, out-of-band manner.

The two mandatory integrity transforms are HMAC-MD5 or HMAC-SHA-1, in each case truncated to 96 bits. Use of AES-CBC MAC is also supported. The recommended confidentiality algorithms are 3-DES in CBC mode, AES in CBC mode, and AES in counter mode.

### 6.1.2 Specifications for Using IPsec

The *Security Extension for UNI and NNI* [Gra03] describes the use of IPsec ESP and IKE to protect control plane traffic (signaling, etc.) between ONEs. If the OAM&P protocol used to access an ONE runs (unicast) Internet Protocol (IP), then using IKE and ESP to protect this access channel is RECOMMENDED[4]. IPsec provides combinations of entity authentication, key management, datagram integrity, replay detection, confidentiality, and security policy management. Security policy management includes establishing security associations and enforcing their proper use.

[4] Even in cases where the OAM&P protocol does not run over IP, it may be possible to protect a portion of the communications path tunneled over an IP network with an IPsec VPN. This particular configuration is out of scope, because it poses no requirements at the ONE.

Because IPsec is perceived to be complex to implement, a major goal of [Gra03] was to specify a simplified profile of IPsec. This section uses [Gra03] to specify how ONEs can protect IP-based OAM&P protocols (command line, web, CORBA, SNMP, or other interfaces running over IP) with the same mechanisms and as few differences as possible. However, in this IA the communications are always between an ONE and a Management System, not between a pair of ONEs, and the protocols being protected are used for OAM&P applications rather than for signaling and services associated with signaling. Implementations protecting OAM&P protocols with IPsec MUST satisfy the requirements in Section 3 of [Gra03] as clarified and modified herein:

- *Section 3.1, Configuration and System Security Issues.*

  Apply this section without any changes or additional specifications.

- *Section 3.2, General Requirements.*

  The exact transport mode selectors are determined by the OAM&P protocol(s) and port(s). The ONE MUST support the ability to limit access so that only permitted traffic is sent over IPsec. It is possible to use a single SA pair in either mode to protect more than one OAM&P protocol. The discussion of discovery protocols does not apply, but the requirements for auditing and using multiple SA pairs do apply.

- *Section 3.3, Transport Mode versus Tunnel Mode.*

  Apply this section without any changes or additional specifications.

- *Section 3.4, DHCP and NAT Traversal.*

  Apply this section without any changes or additional specifications. The connection between ONEs in [Gra03] is a connection between an ONE and a Management System in this IA.

- *Section 3.5, Use of IKE.*

  Apply this section without any changes or additional specifications. The discussion of protecting UNI or NNI applies, in this case, to the applicable OAM&P protocols.

- *Section 3.6. Rekeying.*

  Apply this section without any changes or additional specifications. In addition, note that expired SAs MUST NOT be used. Prior to expiration of an SA, a new SA MUST be established so that the management traffic can be switched over to the new SA prior to the expiration of the original SA. Keep-alive or Hello messages MAY be used for periodic communications to keep the SA from being prematurely torn down due to idleness when management traffic is not being transmitted.

- *Section 3.7, Transforms.*

  Apply this section without any changes or additional specifications.

- *Section 3.8, IPv4 Fragmentation.*

  Apply this section without any changes or additional specifications.

- *Section 3.9, Security Policy Enforcement.*

  Apply this section without any changes or additional specifications.

- *Section 3.10, Naming.*

  Apply this section without any changes or additional specifications.

- *Section 3.11, Authentication.*

  Again, as in [Gra03], if certificates are used, these MUST be machine certificates, not user certificates.

- *Section 3.12, System Issues.*

  Apply this section without any changes or additional specifications.

## 6.2 SSL and TLS

### 6.2.1 SSL and TLS Description

The Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols provide cryptographic authentication, data stream integrity, and data stream confidentiality for TCP connections. For more information, see [Resc01]. SSL and TLS are particularly well suited for protecting http traffic between web browsers and servers, but they may be used to protect any protocol running over TCP (e.g., telnet, rlogin, or even SNMP).

### 6.2.2 Specifications for Using SSL and TLS

In typical e-commerce applications, the 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, authentication of both parties is critical. The recommended 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 less secure alternative method of client authentication is to use a hardware-token-based one-time password system over the 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 (aging, complexity, logging, protection against dictionary attacks, etc., see [T1M1]).

- An ONE or Management System that provides an HTTP server protected by SSL or TLS MUST support SSLv3 with RSA [FCK96]. It MAY also support SSLv3 with DSS and DH and it SHOULD support TLS 1.0 [DA99]. Other protocols (e.g., SSLv2, PCT, and S-HTTP) are outside the scope of this document.
- Clients (e.g., browsers) SHOULD use certificates to authenticate to the server. They MAY, however, use a token-based authentication system or passwords sent over the protected channel.
- Certificates SHOULD be generated with a lifetime of no more than two years. Entire certificate chains MUST be checked for correct names and expiration and SHOULD be checked for revocation.
- Both parties MUST have access to a source of cryptographically strong random or pseudo-random numbers. See [Gut98] and [KSF99] for additional guidelines and recommendations.

- The server MUST support RSA; it MAY support DH-DSS; it MAY support Kerberos as described in [MH99]; and it MAY support the Fortezza cipher suites, but see [Resc01] 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 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.

#### 6.2.2.1 Specifications for Using SSLv3

- An ONE that supports the SSLv3 protocol (protocol version major=3, minor=0) MUST support it as defined in [FCK96].
- Port and protocol selection and use MUST follow [Resc00].
- The ephemeral RSA, anonymous, and Server Gated Cryptography options MUST NOT be used.
- The server MUST use the close_notify alert. The browser SHOULD also use close_notify to complete a two-way closure handshake.
- Both parties SHOULD support protected Rehandshake exchanges.

#### 6.2.2.2 Specifications for Using TLS 1.0

- An ONE that supports the TLS protocol (protocol version is major=3, minor=1) MUST support it as defined in [DA99].
- 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 [Resc00] MUST be followed.
- If TLS 1.0 is supported, the name matching rules specified in [HFPS99] MUST be followed.

- Servers SHOULD and browsers MAY support the use of port numbers described in [KL00].

#### 6.2.2.3 Securing the Browser

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

- Newer browsers (released in 2000 or later) MUST be used instead of older ones, because older protocols like SSLv2 have security defects, and cryptographic strength has increased since the easing of U.S. export restrictions in January 2000. For the same reason, U.S. export-only versions SHOULD NOT be used.
- The browser SHOULD be configured with its security settings to support the specifications 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.

## 6.3 SNMPv3

SNMPv1 and SNMPv2 offer limited security and therefore SNMPv3 may be used as it provides encryption and authentication as part of the core protocol. SNMPv3 with 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 ONE and Management System that uses MIB-based network management running SNMPv3.

### 6.3.1 SNMPv3 over Different Transport Layers

An ONE that supports SNMPv3 access over TCP MUST support:

- SNMPv3 as described below in Section 6.3.2,
- SSL-TLS as described in Section 6.2.1, or
- IPsec as described in Section 6.1.1, which, in this case, is RECOMMENDED.

An ONE that supports SNMP access over UDP MUST support:

- SNMPv3 as described below in Section 6.3.2 or
- IPsec as described in Section 6.2.1, which, in this case, is RECOMMENDED.

An ONE that supports SNMP access over protocols other than TCP and UDP MUST support:

- SNMPv3 as described below in Section 6.3.2.

### 6.3.2 SNMPv3 Description

SNMPv3 is defined in [HPW00], [CHPW00], [LMS00], [WB00], and [WPM00]. 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 have been proposed. For message authentication and data integrity, the SNMPv3 specification lists HMAC-MD5-96 as "shall support" and HMAC-SHA-96 as "should support." 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 only done 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 an ONE 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 caller 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 is not considered to have a requirement covering DoS, because a DoS attack is likely to disrupt all types of communication exchanges, with which the overall security facility, not just that for the management infrastructure, should be concerned and therefore have taken protective and preventive measures against.

### 6.3.3 Specifications for Using MIB-Based Management–SNMPv3

- Implementations MUST support DES-CBC and SHOULD support 3DES-CBC and AES-128-CBC.

- Entities implementing the rekey option MUST have access to a source of cryptographically strong random or pseudo-random numbers. See [Gut98] and [KSF99] 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 Kerberos or IKE MUST NOT use the key localization algorithm option.
- SNMPv3 implementations using the integrity option SHOULD use the timeliness feature.
- Access control lists MAY be used to restrict the IP address from which different SNMP messages are sent.
- SNMP agent logging SHOULD be enabled.

## 6.4 Secure Shell (SSH)

### 6.4.1 SSH Description

The Secure Shell (SSH[5]) defines security protocols that use public key cryptography to establish secure, authenticated sessions between a client and a server.

SSH1 [Yl96] and SSH2 [Car01] are two completely distinct protocols. Both have freely available specifications and have been implemented in freeware and commercial products. Neither is a standard, although SSH2 was described at the time of this writing (November 2002) in several Internet Drafts [Yl02a, Yl02b, Yl02c, Yl02d]. 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 launch a SSH1 server to handle a SSH1 client.

SSH 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 X11 connections or in fact any TCP connection, e.g., to perform a secure remote backup. SSH2 has an explicit capability to secure ftp as well.

A description of SSH begins with the transport layer protocol. In SSH1, two levels of public keys are used. A client sends an authentication request to a server, and the server responds with its long-term public host key and public server key (which changes hourly). In SSH2, only the host key is present. The client compares the host key, which in the former case authenticates the server key, with that which has already been configured. A client may be configured to trust new host keys or not. Note that certificates are not used currently by SSH, but use of a PKI and a device certificate per TNE may be added in the future. 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.

[5] SSH is a registered trademark and Secure Shell is a trademark of SSH Communications Security Ltd. of Finland.

After the client receives and verifies the server's public key(s), it chooses a 256-bit pseudorandom number, which becomes the basic shared secret from which all uni-directional session keys are derived. The random number, a known constant, and the session identifier are double encrypted with the server and host keys in SSH1 or singly encrypted with the host key in SSH2. This value is returned along with a choice of traffic protection algorithms. In SSH1, this provides perfect forward secrecy for the traffic confidentiality keys with respect to the host key.

SSH 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 SSH 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 system. (SSH1 also supports Kerberos for user authentication). 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 SSH encryption, MACs, and secured with end of file messages.

### 6.4.2 Specifications for Using SSH

The following specifications are provided for the use of SSH to protect the management of an ONE.

- Official releases of the software from SSH Communication Security are signed. Implementers or users downloading these releases of SSH SHOULD verify these signatures.
- SSH2 contains improvements in performance, security, and portability over SSH1. In particular, certain active attacks against the SSH1 protocol are prevented in SSH2. Therefore, client and server implementations SHOULD support SSH2.
- Implementations of SSH clients and servers MUST use a cryptographically strong method of generating pseudo-random numbers. See [Gut98] and [KSF99] 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 TNE. Deployments MAY use passwords or, in the case of SSH1, Kerberos. 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 ONEs.
- 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 public key sizes shorter than 768 bits.
- If an ONE 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.

## 6.5 Kerberos

### 6.5.1 Description of Kerberos

Kerberos is a trusted-third party security system that uses a Key Distribution Center (KDC) to establish secure, authenticated sessions between a client and an application server. Kerberos runs at the application layer, so, if Kerberos is used, it requires software support in the applications on the client and the TNE. Each client and each application server have a long-term shared secret established with the KDC. Clients initiate secure communications with Applications Servers by requesting "tickets," which contain the keying material and other credentials needed by the protocol.

### 6.5.2 Specifications for Using Kerberos

An ONE that supports Kerberos MUST implement a Kerberos application server, which accepts Kerberos tickets and authenticators and uses these to provide two-way authentication and to support additional security services. These implementations MUST support the Kerberos Version 5 Specification as defined in [KN93] (see, especially, Section 9.1 in [KN93]). In addition:

- The KDC MUST be physically secured and SHOULD be run on a stand-alone processor with no other applications. It SHOULD have no other users besides Kerberos administrators, and it SHOULD have all other network services disabled (except for logging, auditing, backup, or intrusion detection).
- The KDC MUST turn on Kerberos's auditing.
- The KDC MUST use TCP port 88.
- The KDC SHOULD set all client principals to expire once a year.
- The KDC MUST use a cryptographically strong method of generating random or pseudo-random numbers. See [Gut98] and [KSF99] for additional guidelines and recommendations.
- Cross-realm operation SHOULD be avoided. If cross-realm operation is used, cross-realm authentication MUST be direct.
- TGTs (Ticket Granting Ticket) MUST be issued for at most 8 hours and MUST NOT be renewable for more than seven days.
- Allowable clock skew MUST NOT be more than 5 minutes, and application servers MUST maintain a replay cache of at least 10 minutes. Use of Network Time Protocol [Mil92] is RECOMMENDED.

- Application servers MUST use random passwords and store encrypted passwords in restricted access or otherwise protected files. Application servers SHOULD NOT be allowed to obtain tickets.
- Clients SHOULD NOT use random passwords, unless the clients themselves are implemented as automated scripts, in which case they SHOULD use random passwords and MUST protect these passwords the same way application servers do.
- All principals SHOULD change passwords every 3 to 6 months.
- Clients and application servers MUST support "kerberized" telnet, MUST support "kerberized" ftp, and MAY support "kerberized" rsh or rlogin.
- Clients' tickets SHOULD NOT be "forwardable" and not "proxiable".
- Except for use in automated scripts, tickets MUST NOT be post dated.
- Sessions between clients and application servers MUST use two-way authentication (KRB_AP_REQ MUTUAL REQURED), shall use integrity protection (KRB_SAFE), and may use confidentiality (KRB_PRIV).
- For encryption algorithms, implementations shall support DES-CBC, should support 3-DES-CBC, and should support AES when available.
- For authentication algorithms, implementations shall support MD5, should support SHA-1, may support DES-MAC or DES-MAC-K, shall not use CRC 32, and shall not support MD4.

The following items refer to features currently being considered for IETF standards. If future standards specify such functionality, then:

- Encryption with AES-128-CBC SHOULD be supported.
- Clients MAY be required to use the hardware authentication function.
- Public key initial authentication (PK-INIT) SHOULD be supported.

## 6.6 Other Protocols Supporting Security

### 6.6.1 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.

An ONE 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.

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 [Gut98] and [KSF99] for additional guidelines and recommendations on generating pseudo-random numbers.

### 6.6.2 S/MIME

S/MIME 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.

# 7.0 Objectives Satisfied by Security Systems

Table 2 provides details on which objectives from Section 4 are satisfied by using the security systems in as specified in Sections 6.1 through 6.5. 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 2: Mapping of Objectives to Security Systems.

| **Objective** | **Kerberos** | **SNMPv3** | **SSL-TLS** | **SSH** | **IPsec** |
|---|---|---|---|---|---|
| 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 | | |
|---|---|---|---|---|---|
| A-1 | √ | √ | √ | √ | √ |
| A-2 | √ | √ | √ | √ | √ |
| N-1 | √ | √ | May | May | √ |
| **Objective** | **Kerberos** | **SNMPv3** | **SSL-TLS** | **SSH** | **IPsec** |
| N-2 | Note 5 | Note 6 | √ | √ | √ |
| R-1 | | | | | Note 7 |
| AC-1 | Note 2 | √ | Note 2 | Note 2 | |
| AC-2 | Note 2 | Note 2 | Note 2 | Note 2 | |
| L-1 | May | May | May | May | |
| L-2 | May | May | May | May | |
| L-3 | May | May | May | May | |
| L-4 | | May | May | May | |
| L-5 | May | √ | May | May | May |
| L-6 | √ | √ | √ | | √ |
| T-1 | May | | May | May | √ |
| T-2 | | | | | Note 8 |

Table 2: Mapping of Objectives to Security Systems (Cont.)

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 (Kerberos or IKE) needs to be implemented: Kerberos can satisfy K-1 and K-2, IKE 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: This objective can be satisfied by using the negotiation protocol for GSS-API for the application.

Note 6: N-2 may be satisfied, fully or partially, by using certain key management protocols (e.g., IKE) with SNMPv3.

Note 7: 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 8: The current version of IPsec provides some support for this objective; a newer version, still in draft [April 2003], provides much greater support.

# 8.0 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.


[Bra97] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," IETF RFC 2119, March 1997.

[CHPW00] Case, J., D. Harrington, R. Presuhn, and B. Wijnen, "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)," RFC 3412, Internet Engineering Task Force, December 2002.

[DA99] Dierks, T., and C. Allen, "The TLS Protocol," RFC 2246, Internet Engineering Task Force, January 1999.

[FCK96] Freier, A.O., P. Carlton, and P.C. Kocher, "The SSL Protocol Version 3.0," http://home.netscape.com/eng/ssl3/draft302.txt, November 1996.

[Gra03] Graveman, R., et al., "Security Extension for UNI and NNI," Optical Internetworking Forum Principal Ballot Document, oif2002.373.06, March 25, 2003.

[HC98] Harkins, D., and D. Carrel, "The Internet Key Exchange (IKE)," IETF RFC 2409, November 1998.

[HPW00] Harrington, D., R. Presuhn, and B., Wijnen, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks," RFC 3411, Internet Engineering Task Force, December 2002.

[HFPS99] Housley, R., W. Ford, W. Polk, and D. Solo, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," RFC 3280, Internet Engineering Task Force, April 2002.

[KA98a] Kent, S., and R. Atkinson, "Security Architecture for the Internet Protocol," IETF RFC 2401, November 1998.

[KA98b] Kent, S., and R. Atkinson, "IP Authentication Header," RFC 2402, November 1998.

[KA98c] Kent, S., and R. Atkinson, "IP Encapsulating Security Payload (ESP)," IETF RFC 2406, November 1998.

[KL00] Khare, R., and S. Lawrence, "Upgrading to TLS within HTTP/1.1," RFC 2817, Internet Engineering Task Force, May 2000.

[KN93] Kohl, J., and C. Neuman, "The Kerberos Network Authentication Service (V5)," RFC 1510, Internet Engineering Task Force, September 1993.

[LMS00] Levi, D., P. Meyer, and B. Stewart, “Simple Network Management Protocol (SNMP) Applications,” RFC 3413, Internet Engineering Task Force, December 2002.

[MH99] Medvinsky, A., and M. Hur, “Addition of Kerberos Cipher Suites to Transport Layer Security (TLS),” RFC 2712, Internet Engineering Task Force, October 1999.

[Mil92] Mills, D., “Network Time Protocol (Version 3) Specification, Implementation,” RFC 1305, March 1992. *use this for all of them*

[MSST98] Maughan, D., M. Schertler, M. Schneider, and J. Turner, “Internet Security Association and Key Management Protocol (ISAKMP),” RFC 2408, November 1998.

[Pip98] Piper, D., “The Internet IP Security Domain of Interpretation for ISAKMP,” RFC 2407, November 1998.

[Resc00] Rescorla, E., “HTTP over TLS,” RFC 2818, Internet Engineering Task Force, May 2000.

[Rig00] Rigney, C., et al., “Remote Authentication Dial In User Service (RADIUS),” RFC 2865, Internet Engineering Task Force, June 2000.

[WB00] Wijnen, B., and U. Blumenthal, “User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3),” RFC 3414, Internet Engineering Task Force, December 2002.

[WPM00] Wijnen, B., R. Presuhn, and K. McCloghrie, “View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP),” RFC 3415, Internet Engineering Task Force, December 2002.

[Yl96] Ylönen, T., “SSH—Secure Login Connections over the Internet,” *Proceedings of the Sixth USENIX Security Symposium*, July 1996, pp. 37–42.

[Yl02a] Ylönen, T., et al., “SSH Protocol Architecture,” Internet Draft (work in progress), draft-ietf-secsh-architecture-13.txt, September 20, 2002.

[Yl02b] Ylönen, T., et al., “SSH Transport Protocol,” Internet Draft (work in progress), draft-ietf-secsh-transport-15.txt, September 20, 2002. [Ed. Note: update when issued.]

[Yl02c] Ylönen, T., et al., “SSH Authentication Protocol,” Internet Draft (work in progress), draft-ietf-secsh-userauth-16.txt, September 20, 2002. [Ed. Note: update when issued.]

[Yl02d] Ylönen, T., et al., “SSH Connection Protocol,” Internet Draft (work in progress), draft-ietf-secsh-userauth-16.txt, September 20, 2002. [Ed. Note: update when issued.]

## 8.2 Informative References

[ATMF02] *Methods for Securely Managing ATM Network Elements—Implementation Agreement*, The ATM Forum, AF-SEC-0179.000, April 2002.

[Car01] Carasik, A., "Secure Shell FAQ," Revision 1.4, http://www.tigerlair.com/ssh/faq, February 2001.

[Gut98] Gutmann, P., "Software Generation of Practically Strong Random Numbers," *Seventh USENIX Security Symposium Proceedings*, The USENIX Association, 1998, pp. 243–257.

[IATF] Information Assurance Technical Framework Forum, http://www.iatf.net/protection_profiles/profiles.cfm.

[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.

[NIST01] "Recommendation for Block Cipher Modes of Operation," NIST Special Publication 800-XX, CODEN: NSPUE2, U.S. Government Printing Office, Washington, DC, July 2001.

[OMG02] "Security Services Specification, v1.8," Object Management Group (OMG), March 2002. http://www.omg.org/cgi-bin/doc?formal/02-03-11.pdf.

[Resc01] Rescorla, E., *SSL and TLS*, Addison-Wesley, 2001.

[T1M1] "Baseline Security Requirements for the TNM Management Plane, Draft 3.0," T1M1.5/2000-102, September 2002.

[Tel1] *Generic Requirements for Network Element/Network System Security*, Telcordia GR815, March 2002.

[Tel2] *Generic Requirements for Data Network Security*, Telcordia GR1132, April 1996.