



Security Extension for UNI and NNI

OIF-SEP-01.0

May 8, 2003

Implementation Agreement Created and Approved
by the Optical Internetworking Forum

www.oiforum.com

Working Group: OAM&P

TITLE: Security Extension for UNI and NNI

DATE: May 8, 2003

Document Status: OIF Implementation Agreement OIF-SECEXT

Project Name: Project Plan for Security Contributions

Project Number:

ABSTRACT: This contribution defines a common Security Extension for securing the protocols used in UNI 1.0, UNI 2.0, and NNI. It is based on previously agreed upon security requirements for UNI 2.0 and NNI, which call for a complete, unified, and simplified approach to security. The requirements are reviewed; alternatives are considered; and a recommended approach is selected. Guidelines for using this approach in the most straightforward manner are given, and open issues are identified.

Notice: This implementation agreement 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.

For additional information contact:

The Optical Internetworking Forum, 39355 California Street, Suite 307, Fremont, CA 94538
510-608-5928 phone ☎ info@oiforum.com

Copyright (C) The Optical Internetworking Forum (OIF) (2001). All Rights Reserved.

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.

LIST OF CONTRIBUTORS

Renée Esposito, Booz Allen Hamilton,
esposito_renee@bah.com

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

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

Doug Wiemer, Alcatel,
douglas.wiemer@alcatel.com

TABLE OF CONTENTS

1. INTRODUCTION.....	1
1.1. NNI SECURITY REQUIREMENTS	1
1.2. UNI ENHANCED SECURITY REQUIREMENTS	2
1.3. APPLICABILITY TO UNI 1.0.....	3
1.4. ADDITIONAL REQUIREMENTS	3
1.5. OUT-OF-SCOPE SERVICES	4
1.6. SUMMARY OF REQUIREMENTS	4
2. EVALUATION OF ALTERNATIVE APPROACHES	4
2.1. MULTIPLE APPLICATION LAYER SOLUTIONS	4
2.2. SINGLE APPLICATION LAYER SOLUTION.....	5
2.3. SESSION LAYER SECURITY.....	5
2.4. NETWORK LAYER SECURITY	5
3. GUIDELINES FOR DEPLOYING IPSEC	6
3.1. CONFIGURATION AND SYSTEM SECURITY ISSUES	6
3.2. IPSEC IMPLEMENTATION ARCHITECTURE.....	7
3.3. GENERAL REQUIREMENTS	9
3.4. TRANSPORT MODE VERSUS TUNNEL MODE.....	10
3.5. DHCP AND NAT TRAVERSAL.....	11
3.6. USE OF IKE.....	11
3.7. REKEYING.....	12
3.8. TRANSFORMS	13
3.8.1 Confidentiality Transforms	14
3.8.2 Integrity (Data Origin Authentication) Transforms.....	14
3.8.3 IKE Transforms.....	15
3.9. IPV4 FRAGMENTATION	15
3.10. SECURITY POLICY ENFORCEMENT	16
3.11. NAMING	16
3.12. AUTHENTICATION	17
3.12.1 Machine Certificates.....	17
3.12.2 Pre-Shared Keys	17
3.12.3 IKE and Application-Layer Authentication.....	18
3.13. INTEROPERABILITY GUIDELINES	18
4. OPEN ISSUES, FUTURE DIRECTIONS, AND IMPACTS	19
5. LIST OF ACRONYMS	20
6. ACKNOWLEDGEMENTS.....	22
7. REFERENCES.....	22

LIST OF FIGURES

FIGURE 1: IPSEC/UNI/NNI SCENARIO 1.....	8
FIGURE 2: IPSEC/UNI/NNI SCENARIO 2.....	8

1. Introduction

Section 1 of this IA documents the security requirements from UNI 1.0, the NNI requirements developed by the Carrier WG and other WGs, and the enhanced security requirements (Candidate 12) for UNI 2.0. Security requirements have been identified for NNI and UNI; they are summarized below in Sections 1.1 and 1.2, respectively. Among these requirements are simplicity and commonality, and therefore this Security Extension is written to provide a common set of security mechanisms for UNI and NNI and to show how these can be used as simply as possible. Security mechanisms are required to protect the signaling and routing of optical connections, because these connections carry high volumes of data, involve multiple stakeholders, and consume significant resources. Security mechanisms safeguard transport networks against attacks that may compromise their control planes, seek unauthorized use of their resources, or attempt to gain unauthorized information about their configuration and usage.

For example, a UNI message origin authentication and integrity service can prevent a malicious UNI-C agent from mounting a denial of service attack against a service provider by inserting an excessive number of connection creation requests.

Confidentiality of UNI messages is also likely to be desirable, especially in cases where UNI message attributes include information private to the communicating parties (client and transport network operator). Examples of such attributes include account numbers, contract identification numbers, etc. The case of proxy agent equipment presents increased security and policy control requirements. In this scenario, it is assumed that the UNI-C and UNI-N nodes are connected via networking devices such as layer 2 switches and IP routers. Because these devices could belong to different network operators, might be outside the control of the service provider, and may be publicly accessible, control communication between the UNI-C and UNI-N is subject to increased presence of security threat agents. These threats could mount a variety of attacks, such as:

- IP address spoofing;
- eavesdropping; and
- unauthorized intrusion attempts.

To counter these threats, appropriate security mechanisms have to be employed to protect the UNI signaling and control channel(s).

The goal herein is to specify the optional implementation and deployment of authentication, confidentiality, data integrity, replay detection, and key management services. This Security Extension reduces the number of *different* security measures from what was specified in UNI 1.0, provides options for more extended coverage, defines a common method to secure additional protocols, allows compatibility between UNI and NNI security, and reduces the need for manual intervention. Section 2 examines the different potential security systems considered and explains why one based on IPsec was chosen. Section 3 provides details on how to use IPsec effectively in this application. Finally, Sections 4, 5, and 6 contain open issues and future directions, acknowledgements, and references.

1.1. NNI Security Requirements

Carriers have expressed the need to protect topology, reachability, and addressing details about their own networks. The following excerpts are from the OIF's NNI security requirements:

1. **Guiding Principle:** *The NNI should provide optional mechanisms to (1) authenticate entities exchanging information across an interface; (2) guarantee the integrity of the information (neighbor discovery, service discovery, topology and resource status, signaling) exchanged across an interface; (3) protect the confidentiality of certain types of information [as] required.*
2. **Guiding Principle:** *These mechanisms should protect against both malicious attacks against the optical network as well as unintentionally malfunctioning control entities ... [T]hese mechanisms should be based on a minimal set of comprehensive key management and network or transport layer security solutions. These Guiding Principles are intended to reduce the number of different security measures from what were specified in UNI 1.0 [UNI1], provide options for more extended coverage, include a common method to secure additional protocols, allow compatibility with UNI security, and reduce the need for manual intervention.*
3. **High Priority:** *Network information (except reachability of network clients) shall not be advertised across administrative boundaries. ... Optionally, private network information may be protected with a confidentiality mechanism*
4. **High Priority:** *The signaling network shall be secure; all unauthorized access shall be blocked, and parties may optionally be authenticated*
5. **High Priority:** *The signaling network topology and the signaling node addresses shall not be advertised outside a carrier's domain of trust. Optionally, network topology and node addresses may be protected with a confidentiality mechanism*
6. **High Priority:** *The NNI should provide optional mechanisms for origin authentication, message integrity, and confidentiality of connection management (signaling) messages. ...*
7. *The desired level of security depends, in part, on the types of interface and accounting relation between the two adjacent sub-networks or domains.*

1.2. UNI Enhanced Security Requirements

The security functionality specified in UNI 1.0 [UNI1] is, for the most part, based on what is directly included in the underlying protocols, e.g., LDP [RFC3036] and RSVP [RFC2205, RFC3209, Tsch02]. The UNI 1.0 Implementation Agreement states that this was an expedient choice and was likely to change in subsequent versions of the UNI. (For more details on the security in UNI 1.0, see [UNI1].) The principal shortcomings of the UNI 1.0 approach are:

1. *Incompleteness.* Certain needs like confidentiality and key management are not met.
2. *Inflexibility.* Algorithms, key sizes, and other parameters cannot be adjusted as needs change.
3. *Incompatibility.* Protocol-by-protocol solutions specified in subtly different ways are provided for no technical reason.
4. *Error proneness.* It takes a lot of effort to get security right once, so it is far better to reduce the number of distinct security solutions. This becomes more important as the number of protocols included in the UNI increases.
5. *Unenforceability.* No method exists to enforce what security must be used where.

Therefore, use of a common network or transport layer security system (e.g., SSL, IPsec, or SSH) is needed for this Security Extension. This Security Extension specifies the following:

- A single, optional, security system with a complete set of security services to work across the UNI signaling, routing, discovery, and any other control protocols.
- An *optional* Security Extension to *implement* and to *use*. The goal is to make security *interoperate* when it is implemented and used.

The benefits of this Security Extension are:

1. To move the decision as to what security services to use and how to use them from the protocol designer to the network administrator.
2. To reduce the number of distinct security solutions and therefore the total cost of implementation and deployment.
3. Similarly, to increase the assurance that security is working correctly.
4. To add confidentiality, which may be needed to protect customers' billing and call records and other data including details about users, traffic loads, and network configurations.
5. To add automated key management, which increases interoperability and reduces the likelihood that manual intervention is required to initiate and continue secure operation.
6. To provide a policy-driven solution that allows users to ensure security is working where needed.

1.3. Applicability to UNI 1.0

Because the methods described in this IA can also be applied to a UNI 1.0 compliant interface, this IA also serves as an addendum to UNI 1.0. Note that UNI 1.0 specifies its own security mechanisms for RSVP and LDP, which can be applied to provide certain security properties different from those described in this IA. UNI 1.0 does not, however, preclude the use of any underlying network layer security framework, such as IPsec. Therefore, the methods described in this IA *MAY* be applied to a UNI 1.0 compliant interface. The stated benefit of having a single underlying solution for both NNI and UNI 2.0 is thereby extended to UNI 1.0.

1.4. Additional Requirements

Just as any other functionality, security consumes resources, and it must be designed from a point of view that keeps the costs and benefits properly aligned. Therefore, security should be:

- *Optional to implement and to use*. Some users may decide that they can implement adequate protection by other means (e.g., perimeter access controls and firewalls), so that protocol security is unnecessary. Vendors who choose to serve these users may offer a product without this Security Extension.
- *Interoperable*. However, the purpose of this Security Extension to UNI and NNI is to ensure that protocol security interoperates between vendors' products within carriers' networks. Thus, this Security Extension should contain as few methods, formats, optional features, and algorithms as possible.

- *Synergistic with other functionality in the Optical Network Element (ONE).* This Security Extension for control protocols (including signaling, routing, and discovery) is less costly and less error prone to implement and deploy if it is the same solution used for bearer traffic, management, or user services like VPNs.
- *High assurance.* Solutions should be preferred if they are already well standardized, extensively analyzed, and widely used.
- *Readily available in reference implementations.* This Security Extension encourages the development of complete, interoperable implementations.
- *High quality.* The algorithms and protocols should be chosen first because they are secure. There should be no known defects or serious weaknesses, and the security should be designed to provide secure operation within a broad model of both active and passive attacks.
- *Efficient.* Standard state-of-the-art microprocessors should be able to perform security without undue impact on throughput or latency.

1.5. Out-of-Scope Services

The security services in this IA are aimed to satisfy the above requirements. Therefore, non-repudiation, a stronger form of data origin authentication and data integrity that allows a recipient to convince a third party of these properties, is out of scope. Although non-repudiation may be useful in some cases, practical and efficient mechanisms for non-repudiation are deferred for consideration in future revisions of this IA.

1.6. Summary of Requirements

To satisfy the above requirements, this Security Extension for UNI and NNI control protocols must support confidentiality, data origin authentication, data integrity, and replay detection for all communications. Two-way authentication of the parties must be performed, and this authentication should be integrated with an automated key management system.

2. Evaluation of Alternative Approaches

This section describes four approaches to securing control plane protocols and, after comparing them, chooses the fourth, network layer security, to satisfy the OIF's signaling security requirements. This Section describes various approaches for addressing security for control plane protocols. One way in which these approaches can be differentiated is by the protocol layer at which each operates. In practice, the choice depends on the requirements and the availability of different solutions.

2.1. Multiple Application Layer Solutions

When security is specified protocol by protocol, it usually works differently and often incompatibly in each case. The sets of available security services and algorithms are likely to be incomplete, non-uniform, and non-extensible. This approach makes it difficult to tie security usage to a security policy that enforces what security is required where. When there are more than a couple of protocols, it quickly becomes expensive and error prone. Finally, security for each new protocol has to be considered, *a priori*.

Therefore, it is recommended that this approach not be followed.

2.2. Single Application Layer Solution

Instead of specifying security solutions protocol by protocol, a single security solution could be implemented at the application layer. One standard system for implementing a full set of security services at the application layer is Kerberos (together with several extensions and the GSS-API) [RFC1510, Tung99]. Kerberos provides entity authentication, integrity, and confidentiality services together with key management, so a reasonably complete system of security services is available. As is also the case for protocol by protocol security (Section 2.1), a major drawback to using application layer security is that it has to be implemented within the application code. For Kerberos, in particular, the trusted third party authentication and key management protocols and the traffic protection methods are somewhat restricted and inflexible.

Adopting application-layer security would be a better approach than defining protocol-by-protocol security solutions, but it would also be hard to get it implemented in any legacy code. Therefore, another solution that is easier to implement and deploy is preferable.

2.3. Session Layer Security

Security solutions above the Transport Layer can be implemented easily, yet are limited in their coverage for securing application layers. Two high-quality systems that can be implemented at the Session Layer exist, both of which provide (1) a full set of security services between the TCP layer and the application and (2) public-key-based authentication and key management. One is the Secure Shell (SSH) [Y195, Y196, Cara01, SSH, SSHarch, SSHtrans, SSHauth, SSHcon] and the other is the Secure Sockets Layer or Transport Layer Security (SSL-TLS) [Fre96, RFC2246, RFC2712, Res01] system commonly used to secure Web sessions.

One major advantage to these systems is that they can be inserted between the application and the operating system with minimal difficulty. On the other hand, one drawback to these systems is that SSH and SSL-TLS only secure TCP, not UDP, ICMP, or other protocols. Also, the common ways in which SSL-TLS and SSH are used (client server, with strong authentication requirements on one of the parties) differs from the peer-to-peer model of the optical control plane.

2.4. Network Layer Security

A major advantage of defining security at the network (IPv4 or IPv6) layer is coverage. Security at the network layer can also hide details like higher layer protocol headers and packet sizes. Another advantage of network layer security is that it can be deployed at outboard security processors or intermediate systems like routers or firewalls. All of the protocols mentioned in UNI 1.0 and so far in UNI 2.0 and NNI use an IP network layer. In addition, many important supporting protocols do as well (DNS, NTP, SNMP, etc.). IP Security (IPsec) has all the necessary security functionality as well as coverage for these. IPsec has also been designed to address, to the extent possible, additional security issues like denial of service protection, forward secrecy, and anonymity.

These properties make IPsec the potentially best choice of security solution for UNI and NNI, but two additional points need to be considered:

1. IPsec is regarded as overly complicated. This issue needs to be addressed on multiple fronts. First, the key management protocol in IPsec, IKE, is the source of much of the complexity. It is currently being redesigned with this specific problem in mind, but until that work is completely specified, widely implemented, and generally accepted, a simplified profile for using IKE in a streamlined fashion needs to be provided. Second, the basic IPsec documentation needs to be improved, and the IETF is addressing this as well. Therefore, specific profiles and recommendations are specified in this Security Extension for the OIF's UNI and NNI to simplify the number of choices involved in using IPsec.
2. To specify and enforce a security policy, it is necessary to have a way for the system operator or higher layer protocols to communicate with the IPsec processing layer. One popular method of communicating between IKE and the IPsec layer is PF_KEY described in [RFC2367], but this does not provide a complete solution for general applications.

3. Guidelines for Deploying IPsec

This section describes the Security Extension for protecting optical control plane protocols running over IP networks and defined in the OIF's UNI and NNI. These optical networking control protocols provide functionality that includes signaling, routing, and discovery. This optional Security Extension uses IPsec protocols to provide combinations of authentication, key management, datagram integrity, replay detection, confidentiality, and security policy management. Security policy management includes establishing security associations and enforcing their proper use.

IPsec may be used to secure communications between peer ONEs at the IP layer. The IPsec protocol suite is defined, primarily, in the IP Security Architecture [RFC2401], IKE [RFC2409], IPsec Authentication Header (AH) [RFC2402] and IPsec Encapsulating Security Payload (ESP) [RFC2406] documents. IKE is the key management protocol, whereas AH and ESP are alternative traffic protection protocols. For the purposes of the UNI and NNI Security Extension, AH is not needed and will not be discussed further. ESP provides all the functionality of AH, so using AH, while not prohibited, increases complexity for no gain in security over just using ESP.

An IPsec Security Association (SA) is a one-way association, uniquely identified by a 3-tuple: <SPI, protocol (ESP), destination IP address>*, where SPI stands for Security Parameters Index. The parameters for an IPsec security association are typically established by a key management protocol. These parameters include the encapsulation mode (tunnel or transport), session keys, SPI values, and SA lifetime.

3.1. Configuration and System Security Issues

ONEs are mission-critical facilities that must be isolated and protected (both physically and logically) from security threats. ONEs MAY be protected by security gateways, which shield against attempts to gain improper access or carry out denial of service

* In the currently drafted IETF revisions to IPsec (May 2003), only the SPI and destination address are used to identify a SA.

attacks. This Security Extension SHOULD be configured to leverage the protective services of the existing security infrastructure, including firewall protection.

Whereas this implementation agreement addresses interoperability profiles for using IPsec, it does not address hardening requirements for individual switch implementations. Specifically, systems on such networks should be protected from intrusions, unauthorized use, and unauthorized modification. They should not be configured with unneeded software tools installed or extraneous network connections enabled. Security implementations must be able to protect long term and short term cryptographic keying material from unauthorized access or modification (whether these are pre-shared keys, private keys, master keys, or traffic protection keys). Furthermore, they must be able to preserve their own integrity from unauthorized modifications to their software or configuration, which includes unauthorized modification of public keys used to authenticate other entities or to verify signatures. Finally, security implementations must use unpredictable sources of random numbers for generating keys, nonces (one-time values), initialization vectors, and initial sequence numbers. See, for example, [Gut98, KSN99].

3.2. IPsec Implementation Architecture

The UNI and NNI define an allowable set of Service Invocation Configurations and a set of Signaling Transport Configurations between UNI-C, UNI-N, and NNI devices. The relationship between IPsec and these invocation and transport configurations must be understood in order to describe fully the security environment and solution proposed by this security extension.

Figure 1 demonstrates a possible deployment scenario using the minimum acceptable implementation for UNI compliance. In this case, the Control Channel Realization is provided through an Out-of-Fiber, IP network with In-Fiber Signaling. Control Channel Maintenance using LMP is provided over the Out-of-Fiber, IP network. Note that this diagram demonstrates the simplest case of Service Invocation Configuration options, where the UNI-C and UNI-N agents are providing direct services for the ONEs. As a minimal implementation, it is assumed that the optional Neighbor Discovery and Service Discovery protocols are not used. A single IPsec SA pair is required to secure the UNI Control Channel. Due to the nature of IP networks, it is likely that the Control Channel may need to traverse an intervening Firewall or NAT device. This results in a requirement to use IPsec encapsulated in UDP or IPsec in Tunnel mode with the intervening device providing the IPsec Tunnel endpoint.

Figure 2 demonstrates a more complicated scenario. Security Extensions are deployed in a UNI/NNI network where proxy agents and optional services are used. A proxy agent is used for both the UNI-C and UNI-N. In addition, the UNI option for multiple control channels is demonstrated. The UNI specification considers the Intermediate Signaling Interface (ISI) as out-of-scope and therefore could be another IP Network vulnerable to threat agents. IPsec should be used to secure these connections as well. The result is a scenario in which the IPsec policy may be required to be granular to the UNI/NNI protocol level to map to the IPsec services properly. Control Channel and Neighbor Discovery (Link Verification) protocol traffic are secured in separate IPsec SA pairs.

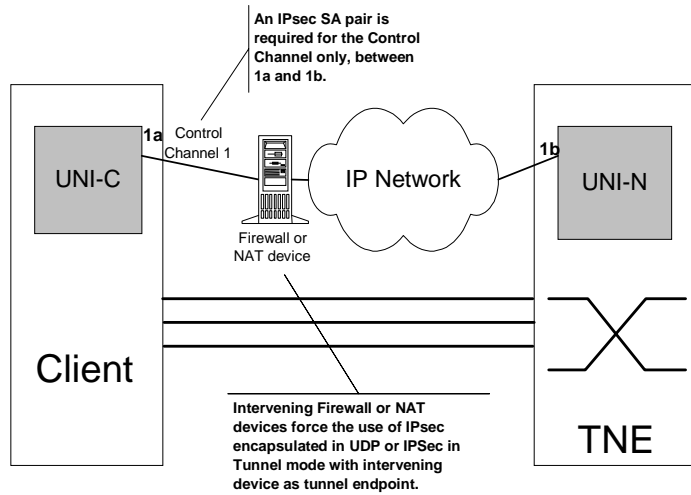


Figure 1: IPsec/UNI/NNI Scenario 1.

Under the UNI/NNI architecture, Client and TNE systems may be deployed using a variety of proxy or direct UNI-C and UNI-N agents. Furthermore, each proxy UNI-C or UNI-N device can serve as the Agent for multiple UNI Clients or TNE systems. This would result in multiple IPsec SA pairs being used for each combination of Client and TNE devices. The implementation selection of direct or proxy agents should be considered when developing systems that support these security extensions. Even though the scenarios described here have been limited to services provided between UNI elements, the same principles apply between NNI elements.

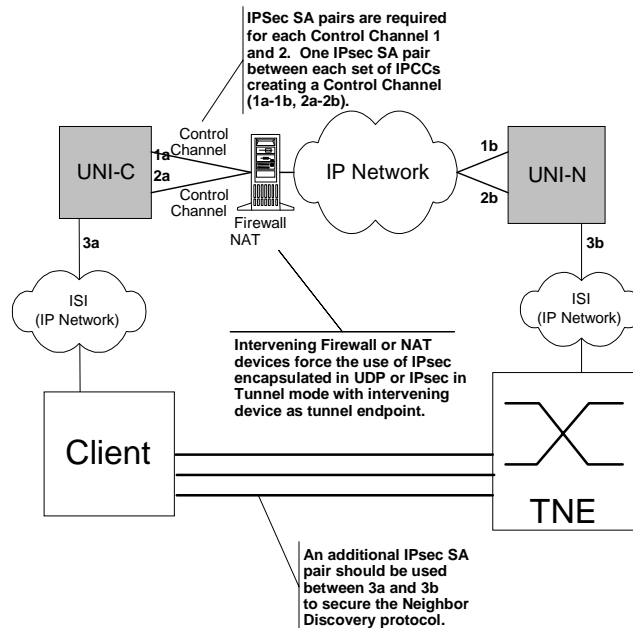


Figure 2: IPsec/UNI/NNI Scenario 2.

3.3. General Requirements

The following general requirements for using IPsec to protect signaling and other control communications between ONEs apply:

1. Unidirectional IPsec ESP SAs are configured in pairs, one in each direction, between communicating ONEs. Between each pair of ONEs using the Security Extension, use of, at a minimum, a single pair of IPsec ESP SAs (one SA in each direction) to protect optical control protocols **MUST** be supported.
2. In the simplest case, in which direct control agents are used and optional signaling protocols are not used, a single pair of SAs is used to protect the control channel between the ONEs. It **MUST** be possible to secure all the optical signaling and control communications with a single IPsec ESP SA pair.
3. ONEs using the Security Extension **MUST** protect the Control Channel traffic, but protection of the Neighbor Discovery and Service Discovery protocol traffic is **OPTIONAL**.
4. The IPsec SAs **MUST** use the Encapsulating Security Payload (ESP) [RFC2406]. Future versions of ESP not specified in RFC 2406 **MAY** be supported.
5. The IPsec ESP SAs in this pair **MUST** use integrity and replay detection with ESP. They **MAY** use confidentiality *in addition*. Therefore, ESP **MUST** support the **NULL** encryption algorithm to allow for integrity and replay detection without confidentiality.
6. ONEs using the Security Extension may decide that some but not all control protocols need confidentiality in addition to integrity and replay detection. In this case, it is **RECOMMENDED** for simplicity that integrity, replay detection, and confidentiality be applied to all these control protocols as in General Requirement (2), above.
7. Applying different qualities of service to datagrams protected by the same SA can result in packets being reordered and falling outside the replay window. In this case, users **SHOULD** enlarge the replay window or avoid different QoS settings. See [RFC2983].
8. Auditing of ESP and IKE (see below) events **MUST** be supported.
9. It may not be possible to secure initial discovery protocol exchanges, so discovery protocols **MAY** run unprotected and **MAY** later establish IPsec ESP SAs and repeat the unprotected communications in protected mode. If discovery protocols can be used to distribute IPsec security policy and configuration information for use with this Security Extension, these protocols constitute a point of attack if they are not secured at least as well as the protocols whose security they configure. Therefore, unprotected discovery mechanisms **MAY** be used to determine where IPsec is available but **MUST NOT** be used for distribution of IPsec security policy and configuration information. In the case of discovery protocols used to distribute security policy or configuration information, at a minimum, per-packet data origin authentication, integrity, and replay protection **MUST** be used to protect the policy and configuration information.
10. ONEs using this Security Extension **MUST NOT** use AH. For the uses of IPsec in this IA (integrity, replay detection, and confidentiality), ESP can do all of these, so offering AH would increase complexity with no gain in functionality.

11. ONEs using this Security Extension SHOULD NOT use data sensitivity labels.
12. ONEs using this Security Extension SHOULD NOT use IP compression.
13. ONEs using this Security Extension MAY apply and MUST accept ESP payload padding. Extended padding options in new versions of ESP SHOULD be accepted.

3.4. Transport Mode versus Tunnel Mode

The main differences between IPsec tunnel mode and transport mode [RFC2401] are discussed below. The bulleted items are explanatory text. The numbered items are requirements.

- *Overhead and MTU.* Tunnel mode introduces an additional IP header into the datagram that results in a corresponding decrease in the path MTU for packets traversing the tunnel and in the maximum segment size of TCP connections running through the tunnel.
 - *Address assignment and configuration.* Using IPsec tunnel mode, it is possible for inner and outer source and destination IP addresses to be different, so that IPsec protection can be applied at security gateways as well as at end systems. For tunnel mode, it is necessary to configure inner and outer address pairs. Alternatively, for transport mode, it is only necessary to configure a single source and destination address pair. IPsec tunnel mode addressing considerations are discussed in detail below.
1. *NAT traversal.* IPsec tunnel mode ESP can traverse NAT in limited circumstances, whereas transport mode ESP cannot traverse NAT. More details are provided below. To enable NAT traversal in the general case, the IPsec NAT traversal functionality described in [UDPIPsec] or [NATIKE] MAY be implemented.
 - *Firewall traversal.* If a protocol traverses multiple administrative domains, firewall administrators may choose to verify the integrity and authenticity of each transiting packet, rather than opening a hole in the firewall for the protocol.
 2. In the case of multiple administrative domains, the endpoints MUST use tunnel mode, not transport mode, to simplify the processing required at the firewall and still support end-to-end IP connectivity.
 3. Conformant OIF NNI Security and OIF UNI Security Extension implementations MUST support ESP [RFC2406] in tunnel mode and MAY support ESP in transport mode.
 4. Tunnel mode MUST be used when security is provided at an intermediate router or firewall or when NAT is performed. This is necessary to meet the requirements of IPsec and NAT.
 - When using tunnel mode, the outer destination address is the address of the ONE peer or the IPsec security gateway used to reach the ONE peer. This address is statically configured or dynamically discovered.
 5. Two mechanisms MAY be used to discover the IPsec security gateway used to reach a particular peer. [RFC2230] defines KX Resource Records (RRs) for IPsec gateway discovery. KX RRs are supported by many DNS server implementations, but they

have not been widely deployed. DNS SRV [RFC2782] can also be used for this purpose.

3.5. DHCP and NAT traversal

When ONEs are deployed within an enterprise's or carrier's network, IP addresses are typically assigned statically. Consequently, support for dynamic IP address assignment is not typically required. As a result, the ONEs that support dynamic address assignment MUST NOT introduce additional security vulnerabilities by doing so.

Using IPsec across NAT services implies extra implementation and configuration effort. Therefore, the network connecting the ONEs that use this Security Extension SHOULD be configured without using NAT. However, if NAT is used, the SAs MAY be terminated at the NAT point, if this provides adequate security along the vulnerable parts of the connection. Otherwise, NAT traversal MUST be performed as described below.

Tunnel mode ESP can traverse NAT in a limited set of circumstances. Tunnel mode ESP may successfully traverse NATs, if (1) there is only one protocol endpoint behind a NAT, (2) "ANY to ANY" selectors are negotiated, and (3) the receiver does not perform source address validation. Before applying this solution, the user should understand the security ramifications of these simplifications.

TCP carried within transport mode ESP cannot traverse NAT, even though ESP itself does not include IP header fields within its message integrity check. To traverse NATs, implementations of this Security Extension SHOULD implement IPsec and IKE NAT traversal, as described in [UDPIPsec] and [NATIKE].

The IKE [NATIKE] and IPsec [UDPIPsec] NAT traversal specifications allow negotiating UDP encapsulation of IPsec if a NAT is detected in the path. By determining the IP address of the NAT, the TCP checksum can be calculated based on a pseudo-header including the NAT-adjusted address and ports. This is done prior to calculating the IPsec message integrity check, and TCP checksum verification will succeed.

3.6. Use of IKE

IKE is an authentication, key exchange, and SA negotiation protocol defined in [RFC2409]. IKE exchanges messages containing payloads defined by ISAKMP [RFC2408]. The IP Security Domain of Interpretation (DOI) [RFC2407] defines two phases, which, together, perform the following functions:

1. Secure cipher suite and options negotiation.
2. Master key generation using, e.g., Diffie-Hellman calculations.
3. Public key or shared-secret-based authentication of end-points.
4. SA management (selector and options negotiation, creation, deletion, and rekeying).

IKE Phase 1 establishes a single, two-way ISAKMP SA. IKE Phase 2 establishes uni-directional ESP SAs, usually in pairs. The session keys for each IPsec ESP SA are derived from a master key by applying cryptographic hash functions. IPsec ESP SAs are uniquely named (currently, see footnote above) by the tuple <Protocol (in this case ESP), destination address, SPI>. A selector, proposed by the IKE Initiator and accepted by the IKE Responder, determines the traffic protected by an IPsec ESP SA. In IPsec ESP transport mode, the IPsec ESP SA selector can be a "filter" or traffic classifier, defined

by a five-tuple: <Source IP address, Destination IP address, transport protocol (UDP/SCTP/TCP), Source port, Destination port>.

IKE defines five types of exchanges (1) Main Mode; (2) New Group Mode; (3) Aggressive Mode; (4) Quick Mode; and (5) Informational Exchange. Main Mode and Aggressive Mode are used only in Phase 1; Quick Mode is used only in Phase 2.

The following profile is specified for using IKE in this Security Extension:

1. The OIF NNI and OIF UNI implementations of this Security Extension **MUST** support IKE [RFC2409] for peer authentication, negotiation of security associations, and key management, using the IPsec DOI [RFC2407].
2. Manual keying of IPsec SAs **MUST NOT** be used, because it does not support rekeying.
3. The OIF NNI and OIF UNI implementations of this Security Extension **MUST** support peer authentication using a pre-shared key, and **SHOULD** support certificate-based peer authentication using digital signatures. To simplify implementations, peer authentication using the public key encryption methods outlined in Sections 5.2 and 5.3 of IKE [RFC2409] **SHOULD NOT** be used.
4. IKE Main Mode with pre-shared key authentication **SHOULD NOT** be used when either of the peers uses a dynamically assigned IP address in situations that would necessitate using a key shared by more than two parties. (For Aggressive Mode, this precaution is not necessary.)
5. When digital signatures are used for authentication, Main Mode **MUST** be supported, Aggressive Mode **MAY** be supported, and New Group Mode **MAY** be supported.
6. When digital signatures are used to achieve authentication, an IKE negotiator **MUST** verify signatures with current credentials and **MUST** check these for revocation. If a valid certificate has not been obtained by other means, the IKE negotiator **MUST** use IKE Certificate Request Payload(s) to specify the certification authority (or authorities) that are trusted in accordance with its local policy. IKE negotiators **MUST** check that the pertinent certificate is currently valid, e.g., by checking the Certificate Revocation List (CRL), before accepting a certificate.
7. To allow service providers to hide the internal structure of their networks, client negotiation, as defined in [RFC2409] **MUST** be supported.
8. The ISAKMP Situation [RFC2407, Section 4.2] **MUST** be SIT_IDENTITY_ONLY.
9. The SA attributes Compress Dictionary Size and Compress Private Algorithm **SHOULD NOT** be used.
10. The Authentication Algorithm **MUST** be used and **MUST NOT** be KDPK.

3.7. Rekeying

IPsec ESP SAs may be used for many protocols and may be long lived. Therefore, implementations of the Security Extension **MUST** provide for rekeying.

Rekeying of an existing IPsec ESP SA pair **MAY** be initiated by either party. For the rekeying process, two new IPsec ESP SAs are created and activated, and the existing IPsec ESP SA pair **MAY** be deleted explicitly. The new outbound SA **SHOULD** be used immediately, and the old inbound SA **SHOULD** be left active for some locally-defined

time, which MAY be different for each party.

If IPsec sequence number extensions as described in [ESpv3] are approved, implementations SHOULD support this mechanism.

All transforms require rekeying when the sequence number space is exhausted, but some may require rekeying more frequently. For example, in CBC-mode, collisions, as examined in [DESANALY], will occur after about $2^{n/2}$ blocks, where n is the block size in bits. It is recommended that rekeying occur before approximately $2^{n/2}$ blocks have been sent on a given SA.

More exactly, [DESANALY] shows that the chance of leakage in CBC mode increases as $O(s^2/2^n)$, where n is the block size in bits, and s is the total number of blocks encrypted. The formula below sets a limit on the number of bytes that can be sent on a CBC SA before rekeying is required:

$$B = (n * 2^{n/2})/8$$

where:

B = maximum bytes sent on the SA, and

n = block size in bits.

This means that cipher block size and key length both need to be considered in the decision to rekey. Triple-DES uses a block size $n = 64$ bits (2^3 bytes); this implies that the SA must be rekeyed before $B = (64 * 2^{32})/8 = 2^{35}$ bytes are sent. If the SA is used for control plane traffic only, traffic rates will be moderate, and this will allow substantial time between rekeying operations.

Note that these particular conclusions do NOT apply to counter mode.

3.8. Transforms

Section 5 of [RFC2406] states:

A compliant ESP implementation MUST support the following mandatory-to-implement algorithms:

- DES in CBC mode
- HMAC with MD5
- HMAC with SHA-1
- NULL Authentication algorithm
- NULL Encryption algorithm

Implementations of this Security Extension MUST support IPsec ESP [RFC2406]. Whenever ESP is used, per-packet data origin integrity and replay detection MUST be used. NULL encryption MUST be supported for SAs using only these two services.

The ability to offer IPsec security services with low processing overhead, low cost, low delay, and high throughput is a major concern. Support for multiple algorithms increases the complexity and expense of hardware design, so one of the goals is to specify a minimal set of sufficiently efficient confidentiality and authentication algorithms.

3.8.1 Confidentiality Transforms

The DES algorithm is specified in [FIPS46-3]; implementation guidelines are found in [FIPS74], and security issues are discussed in [DESDIFF], [DESINT], and [DESCRACK]. The IPsec transform for DES is defined in [RFC2405] and the IPsec transform for 3DES in CBC mode is specified in [RFC2451].

For confidentiality, the ESP mandatory-to-implement algorithm (DES) is unacceptable. As noted in [DESCRACK], DES is vulnerable against modest computational resources and so is inappropriate for use in situations where the issue of confidentiality arises at all. Therefore, to add confidentiality, ESP with 3DES [FIPS46-3] in CBC mode [RFC2451] MUST be supported. DES in CBC mode SHOULD NOT be used. The following transforms MUST NOT be used: ESP_DES_IV64, ESP_DES_IV32, and 3IDEA.

Software implementations of 3DES make excessive computational demands at speeds of 100 Mbps or greater. In addition, 3DES implementations require rekeying prior to exhaustion of the current 32-bit IPsec sequence number space, and thus cannot make use of sequence number extensions, if and when they become available. Therefore, 3DES may be inconvenient or impractical for use at high speeds, especially in software. However, a 3DES IPsec transform has been specified, and hardware is available that runs at 1 Gbps, so using 3DES may be practical for the short term.

In addition to 3DES, the following modes of AES [AES] are defined [NSPUE2]:

- AES in Electronic Codebook (ECB) confidentiality mode,
- AES in Cipher Block Chaining (CBC) confidentiality mode,
- AES in Cipher Feedback (CFB) confidentiality mode,
- AES in Output Feedback (OFB) confidentiality mode,
- AES in Counter (CTR) confidentiality mode,
- AES CBC-MAC authentication mode.

When using AES, it may be advisable to use larger public keys and Diffie-Hellman groups. For further information, see [KeyLen1, KeyLen2]. Additional mod p Diffie-Hellman groups for use with IKE are described in [MODP].

AES in CBC mode [AESCBC] MUST be supported, and AES in counter mode [AESCTR] SHOULD be supported. However, when AES in counter mode is used, it is important to avoid reuse of the counter with the same key (everywhere, forever!). One implication is that it is error prone to use IPsec manual keying with counter mode, which is another reason why manual keying MUST NOT be used.

Another issue with counter mode is that the effect of modifying ciphertext is exactly predictable, which again emphasizes why confidentiality MUST NOT be used without also using message integrity (see section 3.7.2 Integrity Transforms).

3.8.2 Integrity (Data Origin Authentication) Transforms

The MD5 algorithm is specified in [RFC1321]; HMAC is defined in [RFC2104], and security issues with MD5 are discussed in [MD5Attack]. The HMAC-MD5 IPsec transform is specified in [RFC2403] and HMAC-SHA1 in [RFC2404]. For data origin authentication and datagram integrity:

- HMAC-SHA1 and HMAC-MD5 **MUST** be supported,
- AES-CBC-MAC [AESXCBC] with XCBC extensions **SHOULD** be supported, and
- HMAC-SHA-256 truncated to 128 bits [SHAEXT] **MAY** be supported.

HMAC-SHA1 [RFC2404] is preferred over HMAC-MD5, due to concerns that have been raised about the security of MD5 [MD5Attack]. It is most practical to use HMAC-SHA1 as the authentication algorithm in the near future. The HMAC-SHA-256 algorithm [NISTSHA] may be forthcoming, along with an IPsec transform [SHAEXT], so this may merit consideration in the future. AES in CBC-MAC authentication mode with XCBC extensions [AESXCBC] was included because it avoids adding substantial additional code or hardware if AES is already being implemented for confidentiality.

3.8.3 IKE Transforms

IKE transforms are used for securing only IKE traffic (i.e., the ISAKMP SA). That is, these transforms are not used for traffic SAs (including SAs used for protecting routing and signaling protocol messages) established by IKE.

For confidentiality of IKE exchanges, 3DES-CBC and AES-CBC **MUST** be supported; AES in counter mode **MAY** be supported; and DES-CBC **SHOULD NOT** be used.

The MD5 and SHA-1 hash algorithms **MUST** be supported, with SHA-1 preferred, as described in the previous section.

IPSec authentication methods include pre-shared keys, DSS signatures, and RSA signatures. Authentication by RSA decryption **SHOULD NOT** be used. RSA and DSS keys **MUST** be at least 1024 bits and at most 2048 bits.

Master keys for the ISAKMP SA are computed using a Diffie-Hellman group. Implementations **MUST** support the mod p groups of sizes 1024, 1536, and 2048. Groups smaller than 1024 **SHOULD NOT** be used. The elliptic curve groups of sizes 155 and 185 **SHOULD NOT** be used.

3.9. IPv4 Fragmentation

IKE, with certificate-based authentication, can cause IPv4 fragmentation, either with certificate chains or even with a single certificate, if the size of the key or other fields (e.g., the distinguished name and other OIDs) is large enough. Many NATs and firewalls do not handle inbound or outbound fragments properly and may not possess the ability to reassemble the packets. Packet filtering routers also frequently discard fragments after the initial one, because fragments typically do not contain full IP headers that can be compared against an Access Control List.

If IKE fragmentation does not work, the endpoints may be unable to establish a SA. Users **SHOULD** install NAT, firewall, or router code that can properly support fragments. Vendors **SHOULD** advise users, if fragmentation cannot be avoided, to consider:

- Obtaining certificates by other means,
- Reducing the size of the certificate chain,
- Reducing the size of fields like the distinguished name within the certificates.

Fragmentation can also become a concern when prepending IPsec headers to a packet, but Path MTU discovery can reduce this problem. For example, when TCP is used as the transport protocol, then path MTU discovery, described in [RFC1191], [RFC1435], [RFC1981], allows the TCP endpoints to discover the correct MTU, including effects due to IPsec.

However, Path MTU discovery depends on receiving appropriate ICMP messages. IPsec implementations may drop unauthenticated ICMP packets, which results in blackholing in naïve TCP implementations, as described in [RFC2923]. Appropriate TCP behavior is described in section 2.1 of [RFC2923]:

TCP should notice that the connection is timing out. After several timeouts, TCP should attempt to send smaller packets, perhaps turning off the DF flag for each packet. If this succeeds, it should continue to turn off PMTUD for the connection for some reasonable period of time, after which it should probe again to try to determine if the path has changed.

If an IPsec implementation that processes unauthenticated ICMP packets receives an ICMP PMTU, this value should be stored in the SA as proposed in [RFC2401], and IPsec should also notify TCP of this event, so that the new MTU value can be used.

3.10. Security Policy Enforcement

When a connection requiring security is opened, implementations of this Security Extension may wish to check that the connection is protected by IPsec. If policy requires security and IPsec protection is removed on a connection, it **MUST** be reinstated before unprotected IP packets are sent. Because IPsec verifies that each packet arrives on the correct SA, as long as it can be ensured that IPsec protection is in place, then implementations can be assured that each received packet was sent by a trusted ONE peer.

Traffic from one or more than one TCP or UDP protocol **MAY** flow within each IPsec ESP SA. Implementations of the control protocol need not verify that the IP addresses and TCP port values in the packet match the socket information used to set up the connection. This check is performed by IPsec, which prevents malicious entities from sending commands on inappropriate ESP SAs.

3.11. Naming

For IKE implementations, the following fields are used for the Identification Type in the Identification Payload.

The IPsec DOI [RFC2407] defines several types of identification data. In the IKE Phase 1 ID_i and ID_r payloads, implementations running IPv4 **MUST** support the ID_IPV4_ADDR and ID_FQDN Identity Payloads. Implementations running IPv6 stacks **MUST** support the ID_IPV6_ADDR and ID_FQDN Identity Payloads.

The ID_IPV4_ADDR and ID_IPV6_ADDR identities **SHOULD NOT** be used if Aggressive Mode, pre-shared keys, and dynamically assigned IP addresses are used. Other choices such as ID_FQDN **SHOULD** be used.

The ID_USER_FQDN, IP Subnet, IP Address Range, ID_DER_ASN1_DN, and ID_DER_ASN1_GN formats **SHOULD NOT** be used; The ID_KEY_ID Identity

Payload **MUST NOT** be used. As described in [RFC2407], the ID port and protocol fields **MUST** be set to zero or UDP port 500 in Phase 1. Also, as noted in [RFC2407]:

When an IKE exchange is authenticated using certificates (of any format), any IDs used for input to local policy decisions **SHOULD** be contained in the certificate used in the authentication of the exchange.

The Phase 2 Quick Mode exchanges **MUST** explicitly carry the Identity Payload fields (IDci and IDcr). Each Phase 2 IDci and IDcr Payload **SHOULD** carry a single IP address (ID_IPV4_ADDR or ID_IPV6_ADDR) and a single non-zero port number, and **SHOULD NOT** use the IP Subnet or IP Address Range formats. Other ID payload formats **MUST NOT** be used.

3.12. Authentication

Many methods can be used for authentication, including the use of machine certificates, pre-shared keys with IKE, and Application Layer authentication. These three methods are discussed in the following sub-sections.

3.12.1 Machine Certificates

IKE defines both machine and user credentials, but the UNI and NNI IAs have no provision for different users of an ONE. Therefore, the certificate credentials used in IKE negotiation **MUST** be those of a uniquely named machine. The machine certificates are typically issued locally and stored on the Initiator and Target ONEs during an enrollment process.

Security vulnerabilities exist if either the private key corresponding to a certificate is divulged or if an attacker can obtain a signed certificate under false pretenses, so the machine certificate enrollment process **MUST** be strictly controlled.

Smartcard storage of the private key lessens the chance of divulging the private key and provides simple portability of the key. But note, also, that this enables moving a unique key from one machine to another, which may or may not be allowed according to local policy. Local policy may also cover the naming and keying of spares, upgrades, and replacements.

3.12.2 Pre-Shared Keys

In Main Mode the SKEYID_e is used prior to the receipt of the identification payload. Therefore, the selection of the pre-shared key may be based on information contained only in the IP header. However, if dynamic IP address assignment is used, it may not be possible to identify a unique pre-shared key based on the IP address. In this case, the same pre-shared key is usually shared by a group and is no longer able to function as a shared secret between two specific parties. Therefore, neither the Initiator nor Responder identifies itself during IKE Phase 1; it is only known that both parties are members of the group with knowledge of the pre-shared key. This permits any group member to act as a man-in-the-middle. The Responder is not uniquely authenticated unless application-layer two-way authentication is performed. In addition to enabling the attacker to present false data, the attacker may also be able to mount a dictionary attack on legacy authentication methods such as CHAP [RFC1994]. Because this vulnerability is widely present in existing IPsec implementations, Main Mode, pre-shared keys, and dynamic addressing **MUST NOT** be used simultaneously

Group pre-shared keys are not required in Aggressive Mode, because the identity payload is sent earlier in the exchange, and therefore the pre-shared key can be selected based on the ID_FQDN. However, when Aggressive Mode is used the user's identity is exposed, which is often considered undesirable.

Care needs to be taken with IKE Phase 1 Identity Payload selection to allow mapping of identities to pre-shared keys even with Aggressive Mode. If the ID_IPV4_ADDR or ID_IPV6_ADDR Identity Payload is used and addresses are dynamically assigned, mapping of actual identities to keys is not possible. As a result, identities other than ID_IPV4_ADDR and ID_IPV6_ADDR (i.e., ID_FQDN) SHOULD be used whenever Aggressive Mode, pre-shared keys, and dynamically assigned IP addresses are all used.

3.12.3 IKE and Application-Layer Authentication

When the identity (i.e., ID_IPV4_ADDR, ID_IPV6_ADDR, or ID_FQDN) asserted in IKE is authenticated, the resulting derived keys are used to provide per-packet origin authentication, integrity, and replay detection. As a result, the identity verified in the IKE conversation is subsequently verified on reception of each packet.

If machine authentication is used, once an IPsec ESP SA is opened, any process or user on a multi-user machine may be able to send traffic down the SA. This is true for both transport mode and tunnel mode SAs. To limit the potential vulnerability, ONEs with integrated IPsec implementations MUST ensure that "socket access" is appropriately controlled. In the case of an outboard security device, any access from the unprotected ("red") side presents the same type of vulnerability.

3.13. Interoperability Guidelines

One of the goals of this IA is to promote interoperability without requiring manual configuration. This section provides the following guidelines for setting IKE parameters to enhance the likelihood of successful SA negotiation. It also describes how information on security policy configuration can be provided to further enhance the chances of successful SA negotiation.

- *Transform restriction.* Because support for 3DES-CBC and HMAC-SHA1 is required of all implementations, offering these transforms enhances the likelihood of overall successful negotiation. If supported, the AES-CTR [AESCTR], AES-CBC [AESCBC], or AES-XCBC-MAC [AESXCBC] transform combinations may be preferred, with 3DES-CBC and HMAC-SHA1 as a backup offer.
- *Group Restriction.* If 3DES-CBC and HMAC-SHA1 are offered with mod p DH groups, then it is required that a DH group of at least 1024 bits be offered along with it. If the AES suite is the preferred offer, then a mod p DH group of 2048 bits MAY be offered along with it, as noted in [KeyLen1]. If perfect forward secrecy is required in Quick Mode, then the Quick Mode PFS DH group SHOULD be the same as the IKE Phase 1 DH group. This reduces the total number of combinations and enhances the chances for interoperability.
- *Key lifetimes.* If a key lifetime is offered that is longer than desired, it is recommended that the Responder consider the offered lifetime as a maximum, accept it, and then use a Lifetime Notify to inform the other peer of shorter lifetime expiration.

- *Policy configuration.* It may be necessary to configure the security policy of an ONE. This can be done manually or automatically via a security policy distribution mechanism. Alternatively, policy can be supplied via a discovery protocol.
- *Discovery protocols.* It may also be helpful to use a discovery protocol to obtain information about a peer's configuration. While it is generally possible to negotiate security parameters within IKE, some situations with incompatible parameters can cause the IKE negotiation to fail. For instance, the minimum piece of peer configuration information required is whether a connection between ONEs requires IPsec or not. This cannot be determined from the IKE negotiation alone without risking a long timeout, which is undesirable.
- *Main Mode and Aggressive Mode.* IKE negotiation can fail if a mode proposed to a peer is incompatible, so it is helpful to know which modes a peer allows.
- *ESP Mode.* It is legal to propose both transport and tunnel modes in the same offer, but not all IKE implementations support this. Thus, it is useful to know whether a peer prefers tunnel mode or transport mode.
- *Perfect Forward Secrecy (PFS).* It is helpful to know whether a peer allows PFS, because an IKE Phase 2 Quick Mode can fail if an Initiator proposes PFS to a Responder that does not allow it.

4. Open Issues, Future Directions, and Impacts

Four security issues not directly addressed herein may arise in the future:

1. Additional protocols may need security and may not be able to use IPsec. For example, there may be protocols that do not run over IPv4 or IPv6, a separate security system for DNS, new requirements for securing routing protocols, or multicast security issues.
2. As experience is gained using this Security Extension, the OIF should develop guidance and tip sheets addressing configuration, interoperability, and policy issues.
3. This Security Extension provides hop-by-hop, ONE-to-ONE security. End-to-end security for protocols like signaling or link state routing would present new requirements.
4. This Security Extension provides security for the optical control plane. It does not use the control plane to set up security for the user's optical connections. The latter is a completely different problem, which would require a new approach.

Actions in other standards groups, in particular the IETF, may impact the future specifications for securing UNI and NNI. Some of these impacts may include:

- Incorporating future versions of IKE and ESP.
- Adding new cryptographic transforms and groups.
- Monitoring the progress of security consideration recommendations for other protocols, e.g., LMP.
- Pointing out that public-key based methods of key management (including PKI) will likely be used, but these do not require all of the complications of what is marketed

under the banner of PKI. A PKI is not required to use this Security Extension, but formats of widely accepted certificates and other data structures may evolve.

The widespread conversion to IPv6 will change the way this Security Extension is used, but all of the foreseen impacts are favorable, e.g., less use of NAT and no fragmentation.

Impacts of this Security Extension on OIF Implementation Agreements include:

UNI Impact:

Little experience with UNI 1.0 security exists, so the impact of adopting this Security Extension is negligible. Performance needs to be considered, but any reasonable security options can be implemented so that they have minimal performance impact. IPsec has been profiled and simplified to enhance performance. Because this security option is considered a layer below UNI, this proposed UNI security option has no known dependencies on other UNI proposals. Specifically, no special fields in the UNI messages or state machine modifications are required to support this option.

NNI Impact:

This approach satisfies what has been accepted by the Carrier WG as NNI security requirements. A main goal of this Security Extension is to provide completely aligned and therefore lower cost security solutions in UNI and NNI. As in the UNI, the security option is considered a layer below NNI, and it does not require any special support by the NNI routing and signaling protocols.

Other Impact:

Security requirements are also being considered for OAM&P. The considerations for management protocols are slightly different from control protocols, partly because requirements differ and partly because other security systems like SNMPv3, SSH, and SSL exist and are to some extent deployed. Some of the potential security solutions are still being revised and enhanced, so some ongoing coordination with the IETF may be needed.

5. List of Acronyms

3DES	Triple Data Encryption Standard
AES	Advanced Encryption Standard
AH	Authentication Header
CBC	Cipher Block Chaining
CFB	Cipher Feedback
CHAP	Challenge Handshake Authentication Protocol
CRL	Certificate Revocation List
CTR	Counter
DES	Data Encryption Standard
DF	Do not Fragment
DH	Diffie Hellman
DHCP	Dynamic Host Configuration Protocol

DNS	Domain Name System
DOI	Domain of Interpretation
DSS	Digital Signature Standard
ESP	Encapsulating Security Payload
FIPS	Federal Information Processing Standard
Gbps	Gigabits per second
GSS-API	Generic Security Service Application Programming Interface
HMAC	Hashed Message Authentication Code
IA	Implementation Agreement
ICMP	Internet Control Message Protocol
IETF	Internet Engineering Task Force
IKE	Internet Key Exchange
IPsec	Internet Protocol Security
ISAKMP	Internet Security Association and Key Management Protocol
KPDK	Key/Pad/Data/Key
LDP	Label Distribution Protocol
MAC	Message Authentication Code
MD5	Message Digest 5
MTU	Maximum Transmission Unit
NAT	Network Address Translation
NNI	Network to Network Interface
NTP	Network Time Protocol
OFB	Output Feedback
OID	Object Identifier
ONE	Optical Network Element
PFS	Perfect Forward Secrecy
PKI	Public Key Infrastructure
PMTUD	Path Maximum Transmission Unit Discovery
RFC	Request for Comments
RR	Resource Record
RSA	Rivest, Shamir, and Adleman
RSVP	Resource Reservation Protocol
SA	Security Association
SCTP	Stream Control Transmission Protocol
SHA-1	Secure Hash Algorithm
SNMP	Simple Network Management Protocol
SNMPv3	Simple Network Management Protocol Version 3
SPI	Security Parameters Index
SSH	Secure Shell

SSL	Secure Socket Layer
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TNE	Transport Network Element
UDP	User Datagram Protocol
UNI	User to Network Interface
VPN	Virtual Private Network
WG	Working Group
XCBC	Extended Cipher Block Chaining

6. Acknowledgements

Many of the ideas, the outline of topics covered Section 3, and some of the text were borrowed from:

1. B. Aboba et al., "Securing Block Storage Protocols over IP," IETF Work in Progress, draft-ietf-ips-security-16.txt, September 17, 2002.

Some of the ideas for profiling IPsec to protect signaling and routing were taken from:

2. Control Plane Security, ATM Forum Technical Committee Document af-sec-0172.000, November 2001.

The following documents were also helpful:

3. Bellovin, S., "Guidelines for Mandating the Use of IPsec," Internet Draft (work in progress), draft-bellovin-useipsec-00.txt, October 2002.
4. Arkko, J., "Using IPsec to Protect Mobile IPv6 Signaling between Mobile Nodes and Home Agents," Internet Draft (work in progress), draft-ietf-mobileip-mipv6-ha-ipsec-01.txt, October 2002.

Tom Afferton of AT&T, Steve Trowbridge of Lucent Technologies, and Sid Chaudhuri and Dimitrios Pendarakis of Tellium offered helpful comments. Doug Zuckerman of Telcordia Technologies and Jim Jones of Alcatel supported this work in the OIF.

7. References

- [AES] Daemen, J., and V. Rijmen, Advanced Encryption Standard, Rijndael, <http://www.esat.kuleuven.ac.be/~rijmen/rijndael/>. Also available as NIST FIPS Pub 197, November 26, 2001.
- [AESCBC] Frankel, S., S. Kelly, and R. Glenn, "The AES Cipher Algorithm and Its Use With IPsec," Internet Draft (work in progress), draft-ietf-ipsec-ciph-aes-cbc-05.txt, November 2002.
- [AESCTR] Housley, R., "Using AES in Counter Mode with IPsec ESP," Internet Draft (work in progress), draft-ietf-ipsec-ciph-aes-ctr-03.txt, January 2003.
- [AESPERF] Schneier, B., et al., "Performance Comparison of the AES Submissions," <http://www.counterpane.com/AES-performance.html>.
- [AESXCBC] Frankel, S., and H. Herbert, "The AES-XCBC-MAC-96 Algorithm and Its

- Use with IPsec,” Internet Draft (work in progress), draft-ietf-ipsec-ciph-aes-xcbc-mac-04.txt, September 2003.
- [Cara01] Carasik, A., “Secure Shell FAQ,” Revision 1.4, <http://www.tigerlair.com/ssh/faq>, February 2001.
- [CTRMODE] Lipmaa, H., P. Rogaway, and D. Wagner, “CTR-MODE encryption,” Comment on modes of operations, NIST, January 2001. See [MODES].
- [DESANALY] Bellare, M., et al., “A Concrete Treatment of Symmetric Encryption: Analysis of the DES Modes of Operation,” 1997, <http://www.cse.ucsd.edu/users/mihir/>.
- [DESCRACK] Gilmore, J., ed., *Cracking DES: Secrets of Encryption Research, Wiretap Politics, and Chip Design*, O’Reilly & Associates, Sebastapol, CA, 1998.
- [DESDIFF] Biham, E., and A. Shamir, “Differential Cryptanalysis of DES-like Cryptosystems,” *Journal of Cryptology*, Vol. 4, no. 1, pp. 3–72, January 1991.
- [DESINT] Bellare, S., “An Issue With DES-CBC When Used Without Strong Integrity,” *Proceedings of the 32nd IETF*, Danvers, MA, April 1995.
- [DHCPIPsec] Patel, B., et al., “DHCPv4 Configuration of IPsec Tunnel Mode,” Internet Draft (work in progress), draft-ietf-ipsec-dhcp-13.txt, June 2001.
- [DHCPv6] Droms, R., ed., et al., “Dynamic Host Configuration Protocol for IPv6 (DHCPv6),” Internet Draft (work in progress), draft-ietf-dhc-dhcpv6-28.txt, November 2002.
- [ESPv3] Kent, S., “IP Encapsulating Security Payload (ESP),” Internet Draft (work in progress), draft-ietf-ipsec-esp-v3-05.txt, April 2003.
- [FIPS46-3] Data encryption standard (DES), NIST FIPS Pub. 46-3, October 25, 1999.
- [FIPS74] Guidelines for implementing and using the NBS data encryption standard, NIST FIPS Pub. 74, April 1981.
- [Fre96] Freier, A., P. Carlton, and P. Kocher, “The SSL Protocol Version 3.0,” <http://home.netscape.com/eng/ssl3/draft302.txt>, November 1996.
- [Gut98] Gutmann, P., “Software Generation of Practically Strong Random Numbers,” *Seventh USENIX Security Symposium Proceedings*, The USENIX Association, January 1998, pp. 243–257.
- [KSN99] 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.
- [KeyLen1] Orman, H., and P. Hoffman, “Determining Strengths For Public Keys Used For Exchanging Symmetric Keys,” Internet Draft (work in progress), draft-orman-public-key-lengths-05.txt, December 2001.
- [KeyLen2] Lenstra, A., and E. Verheul, “Selecting Cryptographic Key Sizes,” *Journal of Cryptology*, Vol. 14, no. 4, pp. 255–293, 2001.
- [KSF] 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.

- [MD5Attack] Dobbertin, H., "The Status of MD5 After a Recent Attack," *CryptoBytes*, Vol.2 no.2, Summer 1996. Available from www.rsa.com.
- [MODES] "Symmetric Key Block Cipher Modes of Operation," <http://www.nist.gov/modes>.
- [MODP] Kivinen, T., and M. Kojo, "More MODP Diffie-Hellman groups for IKE," Internet Draft (work in progress), draft-ietf-ipsec-ike-modp-groups-05.txt, January 2003.
- [NATIKE] Kivinen, T., et al., "Negotiation of NAT-Traversal in the IKE," Internet Draft (work in progress), draft-ietf-ipsec-nat-t-ike-05.txt, January 2003.
- [NISTSHA] Descriptions of SHA-256, SHA-384, and SHA-512," <http://csrc.nist.gov/cryptval/shs/sha256-384-512.pdf>.
- [NSPUE2] "Recommendation for Block Cipher Modes of Operation," NIST Special Publication 800-XX, CODEN: NSPUE2, U.S. Government Printing Office, Washington, DC, July 2001.
- [Res01] Rescorla, E., *SSL and TLS*, Addison-Wesley, 2001.
- [RFC793] Postel, J., "Transmission Control Protocol," IETF RFC 793, September 1981.
- [RFC1191] Mogul, J., and S. Deering, "Path MTU Discovery," IETF RFC 1191, November 1990.
- [RFC1321] Rivest, R., "The MD5 Message-Digest Algorithm," IETF RFC 1321, April 1992.
- [RFC1435] Knowles, S., "IESG Advice from Experience with Path MTU Discovery," IETF RFC 1435, March 1993.
- [RFC1510] Kohl, J., and C. Neuman, "The Kerberos Network Authentication Service (V5)," IETF RFC 1510, Internet Engineering Task Force, September 1993.
- [RFC1851] Karn, P., P. Metzger, and W. Simpson, "The ESP Triple DES Transform," IETF RFC 1851, September 1995.
- [RFC1981] McCann, J., Deering, S. and J. Mogul, "Path MTU Discovery for IP version 6," IETF RFC 1981, August 1996.
- [RFC1994] Simpson, W., "PPP Challenge Handshake Authentication Protocol (CHAP)," IETF RFC 1994, August 1996.
- [RFC2104] Krawczyk, H., M. Bellare, and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication," IETF RFC 2104, February 1997.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," IETF RFC 2119, March 1997.
- [RFC2131] Droms, R., "Dynamic Host Configuration Protocol," IETF RFC 2131, March 1997.
- [RFC2202] Cheng, P., and R. Glenn, "Test Cases for HMAC-MD5 and HMAC-SHA-1," IETF RFC 2202, September 1997.

- [RFC2205] Braden, R., et al., "Resource ReSerVation Protocol (RSVP) -- Version 1, Functional Specification," IETF RFC 2205, September 1997.
- [RFC2230] Atkinson, R., "Key Exchange Delegation Record for the DNS," IETF RFC 2230, November 1997.
- [RFC2246] Dierks, T., and C. Allen, "The TLS Protocol," IETF RFC 2246, January 1999.
- [RFC2276] Solins, K., "Architectural Principles of Uniform Resource Name Resolution," IETF RFC 2276, Internet Engineering Task Force, January 1998.
- [RFC2367] McDonald, D., C. Metz, and B. Phan, "PF_KEY Key Management API, Version 2," IETF RFC 2367, July 1998.
- [RFC2373] Hinden, R., and S. Deering, "IP Version 6 Addressing Architecture," IETF RFC 2373, July 1998.
- [RFC2401] Kent, S., and R. Atkinson, "Security Architecture for the Internet Protocol," IETF RFC 2401, November 1998.
- [RFC2402] Kent, S., and R. Atkinson, "IP Authentication Header," IETF RFC 2402, November 1998.
- [RFC2403] Madson, C., and R. Glenn, "The Use of HMAC-MD5-96 within ESP and AH," IETF RFC 2403, November 1998.
- [RFC2404] Madson, C., and R. Glenn, "The Use of HMAC-SHA-1-96 within ESP and AH," IETF RFC 2404, November 1998.
- [RFC2405] Madson, C., and N. Doraswamy, "The ESP DES-CBC Cipher Algorithm With Explicit IV," IETF RFC 2405, November 1998.
- [RFC2406] Kent, S., and R. Atkinson, "IP Encapsulating Security Payload (ESP)," IETF RFC 2406, November 1998.
- [RFC2407] Piper, D., "The Internet IP Security Domain of Interpretation for ISAKMP," IETF RFC 2407, November 1998.
- [RFC2408] Maughan, D., M. Schertler, M. Schneider, and J. Turner, "Internet Security Association and Key Management Protocol (ISAKMP)," IETF RFC 2408, November 1998.
- [RFC2409] Harkins, D., and D. Carrel, "The Internet Key Exchange (IKE)," IETF RFC 2409, November 1998.
- [RFC2410] Glenn, R., and S. Kent, "The NULL Encryption Algorithm and Its Use With IPsec," IETF RFC 2410, November 1998.
- [RFC2411] Thayer, R., N. Doraswamy, and R. Glenn, "IP Security Document Roadmap," IETF RFC 2411, November 1998.
- [RFC2412] Orman, H., "The OAKLEY Key Determination Protocol," IETF RFC 2412, November 1998.
- [RFC2451] Pereira, R., and R. Adams, "The ESP CBC-Mode Cipher Algorithms," IETF RFC 2451, November 1998.
- [RFC2608] Guttman, E., C. Perkins, J. Veizades, M. Day, "Service Location Protocol, Version 2," IETF RFC 2608, June 1999.

- [RFC2631] Rescorla, E., "Diffie-Hellman Key Agreement Method," IETF RFC 2631, June 1999.
- [RFC2712] Medvinsky, A., and M. Hur, "Addition of Kerberos Cipher Suites to Transport Layer Security (TLS)," IETF RFC 2712, October 1999.
- [RFC2782] Gulbrandsen, A., "A DNS RR for specifying the location of services (DNS SRV)," IETF RFC 2782, February 2000.
- [RFC2923] Lahey, K., "TCP Problems with Path MTU Discovery," IETF RFC 2923, September 2000.
- [RFC2945] Wu, T., "The SRP Authentication and Key Exchange System," IETF RFC 2945, September 2000.
- [RFC2608] Guttman, E., C. Perkins, J. Veizades, M. Day, "Service Location Protocol, Version 2," IETF RFC 2608, June 1999.
- [RFC2631] Rescorla, E., "Diffie-Hellman Key Agreement Method," IETF RFC 2631, June 1999.
- [RFC2712] Medvinsky, A., and M. Hur, "Addition of Kerberos Cipher Suites to Transport Layer Security (TLS)," IETF RFC 2712, Internet Engineering Task Force, October 1999.
- [RFC2782] Gulbrandsen, A., P. Vixie, and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)," IETF RFC 2782, February 2000.
- [RFC2817] Khare, R., and S. Lawrence, "Upgrading to TLS within HTTP/1.1," IETF RFC 2817, Internet Engineering Task Force, May 2000.
- [RFC2818] Rescorla, E., "HTTP over TLS," IETF RFC 2818, May 2000.
- [RFC2865] Rigney, C., S. Willens, A. Rubens, S. Willens, and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)," IETF RFC 2865, June 2000.
- [RFC2923] Lahey, K., "TCP Problems with Path MTU Discovery," IETF RFC 2923, September 2000.
- [RFC2945] Wu, T., "The SRP Authentication and Key Exchange System," IETF RFC 2945, September 2000.
- [RFC2983] Black, D., "Differentiated Services and Tunnels," IETF RFC 2983, October 2000.
- [RFC3036] Andersson, L., et al., "LDP Specification," IETF RFC 3036, January 2001.
- [RFC3209] Awduche, D., et al., "RSVP-TE: Extensions to RSVP for LSP Tunnels," IETF RFC 3209, December 2001.
- [RFC3280] 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.
- [RFC3411] 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.
- [RFC3412] 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.
- [RFC3413] Levi, D., P. Meyer, and B. Stewart, “Simple Network Management Protocol (SNMP) Applications,” RFC 3413, Internet Engineering Task Force, December 2002.
- [RFC3414] 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.
- [RFC3415] 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.
- [SHAEXT] Frankel, S., and S. Kelly, “The HMAC-SHA-256-128 Algorithm and Its Use With IPsec, “Internet Draft (work in progress), draft-ietf-ipsec-ciph-sha-256-01.txt, June 2002.
- [SRPNDSS] Wu, T., “The Secure Remote Password Protocol,” *Proceedings of the 1998 Internet Society Symposium on Network and Distributed Systems Security*, San Diego, CA, pp. 97-111.
- [SSH] Barrett, D., and R. Silverman, *SSH The Secure Shell: The Definitive Guide*, O’Reilly, 2001.
- [SSHarch] Ylönen, T., et al., “SSH Protocol Architecture,” Internet Draft (work in progress), draft-ietf-secsh-architecture-13.txt, September 20, 2002.
- [SSHtrans] Ylönen, T., et al., “SSH Transport Protocol,” Internet Draft (work in progress), draft-ietf-secsh-transport-15.txt, September 20, 2002.
- [SSHauth] Ylönen, T., et al., “SSH Authentication Protocol,” Internet Draft (work in progress), draft-ietf-secsh-userauth-16.txt, September 20, 2002.
- [SSHcon] Ylönen, T., et al., “SSH Connection Protocol,” Internet Draft (work in progress), draft-ietf-secsh-userauth-16.txt, September 20, 2002.
- [Tsch02] Tschofenig, H., “RSVP Security Properties,” Internet Draft (work in progress), draft-tschofenig-rsvp-sec-properties-00.txt, June 2002.
- [Tung99] Tung, B., *Kerberos, A Network Authentication System*, Addison-Wesley, 1999.
- [UDPIPsec] Huttunen, A., et al., “UDP Encapsulation of IPsec Packets,” Internet Draft (work in progress), draft-ietf-ipsec-udp-encaps-06.txt, January 2003.
- [UNI1] Optical Internetworking Forum, *User Network Interface (UNI) 1.0 Signaling Specification*, Implementation Agreement OIF-UNI-01.1, Oct. 1, 2001.
- [YI95] Ylönen, T., “The SSH (Secure Shell) Remote Login Protocol,” <http://www.tigerlair.com/ssh/faq/ssh1-draft.txt>, November 1995.
- [YI96] Ylönen, T., “SSH—Secure Login Connections over the Internet,” *Proceedings of the Sixth USENIX Security Symposium*, July 1996, pp. 37–42.