

IPS Working Group
INTERNET-DRAFT
Category: Standards Track
<[draft-aboba-ips-iscsi-security-00.txt](#)>
[18](#) August 2001

Bernard Aboba
William Dixon
Microsoft
Joseph Tardo
Uri Elzur
Broadcom
M. Bakke
S. Senum
Cisco Systems
Howard Herbert
Jesse Walker
Intel
J. Satran
Ofer Biran
Charles Kunzinger
IBM
David Black
EMC

Securing iSCSI using IPsec

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/ietf/1id-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at
<http://www.ietf.org/shadow.html>.

Copyright Notice

Copyright (C) The Internet Society (2001). All Rights Reserved.

INTERNET-DRAFT

Securing iSCSI Using IPsec

18 August 2001

Abstract

This document discusses how iSCSI may utilize IPsec to provide authentication, integrity, confidentiality and replay protection.

Table of Contents

1.	Introduction	3
1.1	Terminology	3
1.2	Requirements language	4
2.	iSCSI security requirements	4
2.1	iSCSI security protocol	4
2.2	Rekeying issues	5
2.3	IKE issues	6
2.4	Transform issues	6
3.	iSCSI/IPsec inter-operability guidelines	10
3.1	iSCSI/IPsec binding	10
3.2	Initiating a new iSCSI session	11
3.3	Graceful iSCSI teardown	12
3.4	Non-graceful iSCSI teardown	12
3.5	Fragmentation Issues	13
3.6	Per-packet Security Checks	13
3.7	Application layer CRC	14
3.8	NAT traversal	15
4.	Security considerations	16
4.1	IKE and iSCSI authentication	16
4.2	Certificate authentication	17
4.3	Machine versus user authentication	17
4.4	Pre-shared keys	18
5.	References	19
Appendix A	- Software Performance of IPsec Transforms	23
A.1	Authentication transforms	23
A.2	Encryption and Authentication transforms	26
ACKNOWLEDGMENTS		31
AUTHORS' ADDRESSES		32
Intellectual property statement		34
Full Copyright Statement		34

INTERNET-DRAFT

Securing iSCSI Using IPsec

18 August 2001

[1.](#) Introduction

iSCSI, described in [\[1\]](#), is a connection-oriented command/response protocol. An iSCSI session begins with an iSCSI Initiator connecting to an iSCSI Target over TCP, and performing an iSCSI login. While the iSCSI logon may include mutual authentication of the iSCSI endpoints and negotiation of session parameters, iSCSI does not define its own per-packet authentication, integrity, confidentiality or replay protection mechanisms.

After a successful login, the iSCSI Initiator may issue SCSI commands for execution by the iSCSI Target, which returns a status response for each command, over the same connection. A single connection is used for both command/status messages as well as transfer of data and/or optional command parameters. An iSCSI session may have multiple connections, but a separate login is performed on each. The iSCSI session terminates when its last connection is closed.

IPsec is a protocol suite which is used to secure communication at the network layer between two peers. The IPsec protocol suite is specified within the IP Security Architecture [\[6\]](#), IKE [\[7\]](#), IPsec Authentication Header (AH) [\[3\]](#) and IPsec Encapsulating Security Payload (ESP) [\[4\]](#) documents. IKE is the key management protocol while AH and ESP are used to protect IP traffic.

This draft proposes use of the IPsec protocol suite for protecting iSCSI traffic over IP networks, and discusses how IPsec and iSCSI should be used together.

[1.1.](#) Terminology

iSCSI iSCSI is a client-server protocol in which clients (Initiators) open connections to servers (Targets).

Initiator The iSCSI Initiator connects to the Target on well-known TCP port <TBD>. The iSCSI Initiator then issues SCSI commands for

execution by the iSCSI Target.

Target The iSCSI Target listens on a well-known TCP port for incoming connections, and returns a status response for each command issued by the iSCSI Initiator, over the same connection.

[1.2.](#) Requirements language

In this document, the key words "MAY", "MUST", "MUST NOT", "optional", "recommended", "SHOULD", and "SHOULD NOT", are to be interpreted as described in [\[2\]](#).

[2.](#) iSCSI security requirements

The iSCSI protocol is used to transmit SCSI commands over IP networks. Therefore, both the control and data packets of iSCSI are vulnerable to attack. Examples of attacks include:

- [1] An adversary may try to discover user identities by snooping data packets.
- [2] An adversary may try to modify packets (both control and data).
- [3] An adversary may try to hijack the iSCSI connection.
- [4] An adversary can launch denial of service attacks by terminating iSCSI connections, such as by sending a TCP reset.
- [5] An adversary may attempt to disrupt the iSCSI logon negotiation so as to weaken the iSCSI authentication process or gain access to user passwords.

To address these threats, the iSCSI security protocol **MUST** provide authentication, integrity and replay protection for control and data packets. It **MUST** provide confidentiality for control and data packets.

An iSCSI security protocol MUST also provide a scalable approach to key management.

The iSCSI protocol, and iSCSI logon authentication do not meet the security requirements for iSCSI. iSCSI logon authentication provides mutual authentication between the iSCSI Initiator and Target at connection origination, but does not protect control and data traffic on a per packet basis, leaving the iSCSI connection vulnerable to attack. iSCSI logon authentication via SRP [48] mutually authenticates the Initiator to the Target, but does not by itself provide per-packet authentication, integrity, confidentiality or replay protection. In addition, iSCSI logon authentication, outlined in [1], does not provide for a protected ciphersuite negotiation. Therefore, iSCSI logon provides a weak security solution.

[2.1.](#) iSCSI Security Protocol

All iSCSI security compliant implementations MUST implement IPsec ESP in transport mode for securing both iSCSI control and data packets. If

confidentiality is not used (e.g., iSCSI data traffic), ESP with NULL encryption may be used. The implementations MUST implement replay protection mechanisms of IPsec.

iSCSI security MUST meet the key management requirements of the IPsec protocol suite. IKE SHOULD be supported for authentication, security association negotiation, and key management using the IPsec DOI [5].

To provide authentication, integrity and replay protection of iSCSI PDUs, iSCSI security implementations MUST support transport mode ESP with NULL encryption and HMAC-SHA1 authentication. Transport mode ESP with AES in OCB mode MUST be supported to provide confidentiality as well as authentication, integrity and replay protection.

[2.2.](#) Rekeying issues

It is expected that iSCSI implementations will need to operate at high speed. For example, implementations operating at 1 Gbps are currently in design, with 10 Gbps implementations to follow shortly thereafter. At these speeds, iSCSI will rapidly cycle through the 32-bit IPsec sequence number space.

For example, a 1 Gbps implementation sending 64 octet packets exclusively would exhaust the 32-bit sequence number space in 2200 seconds, or 37 minutes. With 1518 octet packets, exhaustion would occur in 14.5 hours.

A 10 Gbs implementation sending 64 octet packets would exhaust the sequence number space in 220 seconds or 3.67 minutes. With 1518 octet packets, this would occur within 1.45 hours.

As a result, iSCSI implementations operating at speeds of 1 Gbps or less MAY implement the IPsec sequence number extension, described in [49]. 10 Gbps or faster implementations SHOULD implement the extension specification.

Note that depending on the transform in use, it is possible that rekeying will be required prior to exhaustion of the sequence number space. Bellare et. al. have formalized this in [51], showing that the insecurity of 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.

This formula sets a limit on the bytes that can be sent on a CBC SA before a rekey is required:

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

Where:

B = maximum bytes sent on the SA
 n = block size in bits

This means that cipher block size as well as key length need to be considered in the rekey decision. 3DES uses a block size $n = 64$ bits (2^3 bytes); this implies that the SA must be rekeyed before $B = (64/8) * (2^{32}) = 2^{35}$ bytes are sent. At 1 Gbps, this implies that a rekey will be required every 274.9 seconds (4.6 minutes); at 10 Gbps, a rekey is required every 27.5 seconds. In practice, a safety margin is required so the required rekey times will be even smaller.

In terms of the sequence number space, for a 3DES encrypted message of 512 = 2^9 bytes (2^6 blocks) this implies that the key has become insecure after about 2^{26} messages. This is $s = 2^{26} * 2^6 = 2^{32}$

blocks and $(2^{32})^2/2^{64} = 1$. With the 3DES cipher in CBC mode, it would be prudent to rekey more often, such as every 2^{20} messages or 2^{29} bytes. This would imply a rekey time of 4.29 seconds at 1 Gbps or [0.43](#) seconds at 10 Gbps. These exceedingly short rekey times make it very difficult to utilize 3DES effectively to secure iSCSI.

In comparison, AES-CBC uses a block size of 128 bits (2^4 bytes). This enables $B = (128/8) * (2^{64}) = 2^{68}$ bytes to be sent prior to requiring a rekey. This means that the required rekey times are 2^{33} times longer than for 3DES.

In terms of the sequence number space, for an AES encrypted message of [512](#) = 2^9 bytes (2^5 blocks) this implies that the key has become insecure after about 2^{59} messages ($2^{59} * 2^5)^2/2^{128} = 1$. This means that the entire current ESP sequence space of 2^{32} messages could be consumed without compromising the key. AES would still permit a safe CBC mode construction if the ESP sequence space were expanded to 48 bits, since $(2^{48} * 2^5)^2/2^{128} = 2^{-22}$.

[2.3](#). IKE issues

As noted in [\[48\]](#), there are situations where it is necessary for IKE to be implemented in firmware. With the proliferation of IPsec host implementations, these issues are most likely to arise in Target designs.

In such situations, it is important to keep the size of the IKE implementation within strict limits. As noted in [\[48\]](#) an upper bound on the size of an IKE implementation might be considered to be 800 KB, with [80](#) KB enabling implementation in a wide range of situations.

As noted in Table 1 on the next page, IKE implementations currently exist which meet the requirements. Therefore, while removal of seldomly

used IKE functionality (such as the nonce authentication methods) would reduce complexity, iSCSI implementations typically will not require this in order to fit within the code size budget.

[2.4](#). Transform issues

Since iSCSI implementations may operate at speeds of 1 Gbps or greater, the ability to offer IPsec security services at high speeds is of

intense concern. Since support for multiple algorithms multiplies the complexity and expense of hardware design, one of the goals of the transform selection effort is to find a minimal set of confidentiality and authentication algorithms implementable in hardware at speeds of up to 10 Gbps, as well as being efficient for implementation in software at speeds of 100 Mbps.

In this specification, we primarily concern ourselves with IPsec transforms that have already been specified, and for which parts are available that can run at 1 Gbps line rate. Where existing algorithms do not gracefully scale to 10 Gbps, we further consider algorithms for which transform specifications are not yet complete, but for which parts are expected to be available for inclusion in products shipping within the next 12 months. As the state of the art advances, the range of feasible algorithms will broaden and additional mandatory-to-implement algorithms may be considered.

This draft proposes that iSCSI security utilize IPsec transport mode ESP. [Section 5 of RFC 2406](#) [4] 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

"

The DES algorithm is specified in [29]; implementation guidelines are found in [30], and security issues are discussed in [31],[43], [17]. The DES IPsec transform is defined in [32] and the 3DES in CBC mode IPsec transform is specified in [33].

The MD5 algorithm is specified in [8]; HMAC is defined in [19] and security issues with MD5 are discussed in [19]. The HMAC-MD5 IPsec transform is specified in [24]. The HMAC-SHA1 IPsec transform is specified in [25].

the following modes [37] of AES, defined in [34],[35]:

- 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

The Modes [36] effort is also considering a number of additional algorithms, including the following:

PMAC

HMAC-SHA1 [25] is to be preferred to HMAC-MD5, due to concerns that have been raised about the security of MD5 [9]. HMAC-SHA1 parts are currently available that run at 1 Gbps, the algorithm is implementable in hardware at speeds up to 10 Gbps, and an IPsec transform specification [25] exists. As a result, it is most practical to utilize HMAC-SHA1 as the authentication algorithm for products shipping in the near future. As a result, iSCSI security implementations MUST implement HMAC with SHA1.

The HMAC-SHA2 algorithm [28] is also under development, including an IPsec transform [45], so that this may merit consideration in the future. Authentication transforms also exist that are considerably more efficient to implement than either HMAC-SHA1 or HMAC-SHA2. UMAC [27],[44] is more efficient to implement in software and PMAC [26] is more efficient to implement in hardware. PMAC is currently being evaluated as part of the NIST modes effort [36] but an IPsec transform does not yet exist; neither does a UMAC transform.

For confidentiality, the ESP mandatory-to-implement algorithm (DES) is unacceptable for use with iSCSI security. As noted in [17], DES is crackable with modest computation resources, and so is inappropriate for use in situations requiring high levels of security. 3DES also has significant disadvantages. As described in [Appendix A](#), 3DES software implementations make excessive computational demands at speeds of 100 Mbps or greater, effectively ruling out software-only iSCSI implementations at speeds of 100 Mbps or less.

In addition, 3DES implementations require rekeying prior to exhaustion of the current 32-bit IPsec sequence number space, and thus would not be able to make use of sequence space extensions, if they were available. This means that 3DES will require very frequent rekeying at speeds of 10 Gbps or greater.

For these reasons, while hardware implementations of 3DES are available at the required speeds, and IPsec transforms are available, 3DES is inconvenient for use at high speeds, as well as being impractical for implementation in software at slower speeds (100 Mbps). As a result, 3DES is optional for use with iSCSI security.

Implementation	Code size (octets)	Release
Pluto (FreeSWAN)	258KB	Linux FreeSWAN x86
Racoon (KAME)	400KB	NetBSD 1.5 x86
Isakmpd (Erickson)	283KB	NetBSD 1.5 x86
WindRiver	142KB	PowerPC
Cisco VPN1700	222KB	PowerPC
Cisco VPN3000	350K	PowerPC
Cisco VPN7200	228KB	MIPS

Table 1 - Code Size for IKE implementations

INTERNET-DRAFT

Securing iSCSI Using IPsec

18 August 2001

iSCSI security implementations MUST implement AES in OCB mode; an IPsec transform for this does not yet exist.

[3.](#) iSCSI/IPsec inter-operability guidelines

The following guidelines are established to meet iSCSI security requirements using IPsec in practical situations.

[3.1.](#) iSCSI/IPsec binding

An iSCSI session [[1](#)], comprised of one or more TCP connections, is identified by the 2-tuple of the Initiator-defined identifier and the Target-defined identifier, <ISID, TSID>. Each connection within a given session is assigned a unique Connection Identification, CID. The TCP connection is identified by the 5-tuple <Source IP address, Destination IP address, TCP, Source Port, Destination Port>. An IPsec Phase 2 SA is identified by the 3-tuple <ESP, destination address, SPI>.

The iSCSI session and connection information is carried within the iSCSI Login Commands, transported over TCP. Since an iSCSI initiator may have multiple interfaces, iSCSI connections within an iSCSI session may be initiated from different IP addresses. Similarly, multiple iSCSI targets may exist behind a single IP address, so that there may be multiple iSCSI sessions between a given <source IP address, destination IP address> pair.

The relationship between iSCSI sessions, TCP connections and IKE Phase 1 and Phase 2 SAs is as follows:

- [1] An iSCSI initiator or target may have more than one interface, and therefore may have multiple IP addresses. Also, multiple iSCSI initiators and targets may exist behind a single IP address. As a result, an iSCSI Session may correspond to multiple IKE Phase 1 Security Associations, though typically a single IKE Phase 1 security association will exist for an <Initiator IP address, Target IP address> tuple.
- [2] Each TCP connection within an iSCSI Session is protected by a separate IKE Phase 2 SA, with descriptors specific to that TCP

connection. Each IKE Phase 2 SA protects only a single TCP connection, and each TCP connection is transported under only one IKE Phase 2 SA.

Given this, all the information needed for the iSCSI/IPsec binding is contained within the iSCSI Login messages from the iSCSI Initiator and Target. This includes the binding between an IKE Phase 1 SA and the corresponding iSCSI sessions, as well as the binding between an IPsec Phase 2 SA and the TCP connection and iSCSI connection ID.

[3.2.](#) Initiating a New iSCSI Session

In order to create a new iSCSI Session, an Initiator implementing iSCSI security first establishes IKE Phase 1 and Phase 2 SAs, then exchanges iSCSI control messages over an IPsec-secured TCP connection. The iSCSI Initiator contacts the Target on well-known TCP port <TBD>. The Initiator and Target IKE implementation MUST successfully complete the IKE phase 1 and Phase 2 negotiations before the initial TCP connection setup messages are exchanged so that these messages can be IPsec protected. From this point forward, subsequent iSCSI connections established within the iSCSI session will be protected by IKE Phase 2 SAs derived from the IKE Phase 1 SA.

In the Phase 2 Quick Mode exchanges used to protected individual iSCSI connections, the Identity Payload fields MUST be present. These fields carry the source and destination addresses and source and destination ports of the iSCSI Initiator and Targets, thus binding the Phase 2 security association to specific TCP and iSCSI connections. The IKE Quick Mode ID payloads MUST carry individual addresses, and MUST NOT use the IP Subnet or IP Address Range formats.

Once the IKE Phase 2 negotiations are complete and the TCP connection is established over IPsec, the iSCSI Initiator MUST send the iSCSI Login command over the TCP connection secured by the recently negotiated Quick Mode SA.

The Initiator fills in the ISID field, and leaves the TSID field set to zero, to indicate that it is the first message of a new session establishment exchange. The Initiator also fills in a CID value, which is associated with the iSCSI connection corresponding to the the TCP connection secured by the Quick Mode SA. When the iSCSI Target replies with its Login Command, both iSCSI devices will know the TSID, and

therefore the iSCSI session identifier <ISDI, TSID>.

At this point, a binding is established between the iSCSI session identifier and the IKE Phase 1 SAs. A single iSCSI session identifier may have multiple associated IKE Phase 1 SAs, and each IKE Phase 1 SA may have multiple associated iSCSI session identifiers. In addition, a binding is established between the iSCSI connection identifier CID, the TCP connection 5-tuple, and the IPsec Phase 2 SA, as identified by the <ESP, destination address, SPI> combination. Each iSCSI connection corresponds to a single TCP connection and IPsec Phase 2 SA.

Before adding a new connection to an existing iSCSI Session, a new IKE Quick Mode exchange MUST occur, under the protection of an IKE Phase 1 SA.

Within IKE, each key refresh requires that a new security association be established. In practice there is a time interval during which an old, about-to-expire SA and newly established SA will both be valid. The IPsec implementation will choose which security association to use based on local policy, and iSCSI concerns play no role in this selection process.

[3.3.](#) Graceful iSCSI Teardown

Mechanisms within iSCSI provide for both graceful and non-graceful teardown of iSCSI Sessions or individual TCP connections within a given session. The iSCSI Logout command is used to effect graceful teardown. This command allows the iSCSI Initiator to request that:

- [a] the session be closed
- [b] a specific connection within the session be closed
- [c] a specific connection be marked for recovery, or
- [d] a specific connection be closed at the Target's request.

When the iSCSI implementation wishes to close a session, it MUST use the appropriate iSCSI commands to accomplish this. After exchanging the appropriate iSCSI control messages for session closure, the iSCSI

security implementation SHOULD initiate a half-close of each TCP connection within the iSCSI session. Since a given IKE Phase 1 SA may be bound to multiple iSCSI sessions, the iSCSI implementation will only delete the IKE Phase 1 SAs bound to the iSCSI session if there are no remaining iSCSI sessions bound to those SAs. For those Phase 1 SAs that are deleted, the iSCSI security implementation will also delete the IKE Phase 2 SAs bound to them.

When the iSCSI security implementation wishes to close an individual TCP connection while leaving the parent iSCSI session active, it SHOULD half-close the TCP connection. This results in a FIN being sent, putting the TCP connection into the FIN WAIT-1 state, as described in [10]. After the other side responds, the TIME WAIT state is entered. After the expiration of the TIME WAIT timeout, the IKE Phase 2 security association bound to the TCP connection MUST be closed. Closing the TCP connection prior to deleting the IKE Phase 2 SA ensures that all the TCP packets sent on the connection are IPsec-protected.

[3.4.](#) Non-graceful iSCSI Teardown

If the iSCSI security implementation becomes aware that a given TCP connection has unexpectedly failed, it SHOULD delete the associated IKE Phase 2 security association. If the IKE implementation receives a

Phase 2 Delete message for a security association bound to a TCP connection, it SHOULD notify the iSCSI security implementation. If the TCP connection whose SA was deleted is one which a Logout Command/Logout Response sequence marked for removal from the iSCSI session, then the IKE Phase 2 Delete message serves as confirmation that the iSCSI peer has executed an iSCSI teardown process for the connection. The iSCSI connection state and any associated filters can now be safely removed.

[Issue: If a Logout Command/Logout Response was not received, then what do we do?]

If an IKE implementation receives a Phase 1 Delete message for a Phase 1 Security Association bound to one or more iSCSI sessions, then it SHOULD notify the iSCSI security implementation. It also SHOULD delete the associated IKE Phase 2 security associations.

[3.5.](#) Fragmentation Issues

Fragmentation can become a concern when prepending IPsec headers to an iSCSI frame. One mechanism which can be used to reduce this problem is to utilize path MTU discovery within the iSCSI transport protocol. For example, if TCP is used as the iSCSI transport, then path MTU discovery [11]-[13], can be used to enable the TCP endpoints to discover the correct MTU, including effects due to IPsec encapsulation.

However, Path MTU discovery fails when appropriate ICMP messages are not received by the host. This occurs in IPsec implementations which drop unauthenticated ICMP packets. This can result in blackholing in naive TCP implementations, as described in [14]. Appropriate TCP behavior is described in section 2.1 of [14]:

"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 ICMP PMTU is received by an IPsec implementation that processes unauthenticated ICMP packets, this value SHOULD be stored in the SA as proposed in [6], and IPsec should also provide notification of this event to TCP so that the new MTU value can be correctly reflected.

[3.6.](#) Per-packet Security Checks

When a packet arrives from a connection which requires security, iSCSI MUST check to ensure that the packet was decrypted and/or authenticated by IPsec. Since IPsec already verifies that the packet arrived in the

correct SA, iSCSI can be assured that the packet was indeed sent by a trusted peer.

When used with iSCSI, IPsec will negotiate a separate Phase 2 SA for each TCP connection, with IPsec filters specific to the TCP connection. As a result, only traffic for a single TCP connection will flow within each IPsec Phase 2 SA. iSCSI security implementations need not verify that the IP addresses and TCP port values in the packet match the socket information which was used to setup the iSCSI connection. This check will be performed by IPsec, preventing malicious peers from sending iSCSI commands on inappropriate Quick Mode SAs.

[3.7.](#) Application-layer CRC

iSCSI's error detection and recovery assumes that the TCP and IP checksums provide inadequate integrity protection and hence incorporates [32](#) bit CRCs to protect its headers and data. When a receiver CRC check fails (i.e., CRC computed at receiver does not match the received CRC), all data covered by that CRC must be discarded. Since presumably the error was not detected by the TCP checksum, TCP retransmission will not occur and thus cannot assist in recovering from the error. iSCSI contains both data and command retry mechanisms to deal with the resulting situations, including SNACK, the ability to reissue R2T commands, and the retry (X) bit for commands.

IPsec per-packet authentication and integrity protection offers strong protection against an attacker attempting to modify packets in transit, as well as unintentional packet modifications caused by router malfunctions. This protection is considerably stronger than both the 16-bit TCP checksum [[11](#)] and the 32-bit application-layer CRC that has been proposed for use with iSCSI [[1](#)]. Since IPsec integrity protection occurs below TCP, if an error is discovered, then the packet will be discarded and TCP retransmission will occur, so that no recovery action need be taken at the iSCSI layer.

As a result, if end-to-end IPsec integrity protection is known to be in place, and covers the entire connection between iSCSI endpoints (or the portion thereof that requires this additional integrity connection), portions of iSCSI can be simplified. In this case, the iSCSI CRC and mechanisms to recover from CRC check failures are not necessary. If the iSCSI CRC is negotiated, the recovery logic SHOULD be simplified to regard any CRC check failure as fatal (e.g., generate a SCSI CHECK CONDITION on data error, close the corresponding TCP connection on header error) because it will be rare for errors undetected by IPsec integrity protection to be detected by the iSCSI CRC.

Note that omitting the iSCSI CRC is not advisable in all situations where IPsec integrity protection is employed. When IPsec, TCP and iSCSI

are implemented purely in software, it can be argued that additional failure modes may be detected by the TCP checksum and/or iSCSI CRC, and therefore that these additional checks are worthwhile. For example, verification of the cryptographic message integrity check might be

successful, but then after the segment is copied as part of TCP processing, a memory error might cause TCP checksum or iSCSI CRC verification to fail.

Given the demand for high speed iSCSI security implementations, implementations utilizing hardware offload are expected to become common. Where IPsec processing as well as TCP checksum and iSCSI CRC verification are offloaded within the NIC, these individual checks no longer provide diversity against single points of failure. Since both the IPsec cryptographic message integrity check, the TCP checksum and the application layer CRC will have been verified prior to transferring data across the bus, subsequent transfer or memory errors will not be detected.

As a result, where iSCSI security is supported, and IPsec processing is offloaded to the NIC, the iSCSI CRC is not necessary and the implementation may not request it. There are two exceptions to this:

- [1] If an implementation is an iSCSI-iSCSI proxy or gateway, it can propagate the iSCSI data CRC from one iSCSI connection to another. In this case, the iSCSI CRC is useful to protect iSCSI data against memory, bus, or software errors within the proxy or gateway, and requesting it is desirable.
- [2] If IPsec is provided by a device external to the actual iSCSI device, the iSCSI header and data CRCs can be kept across the part of the connection that is not protected by IPsec. For instance, the iSCSI connection could traverse an extra bus, interface card, network, interface card, and bus between the iSCSI device and the device providing IPsec. In this case, the iSCSI CRC is desirable, and the iSCSI implementation behind the IPsec device may request it.

Note that if both ends of the connection are on the same segment, then traffic will be effectively protected by the layer 2 CRC, so that negotiation of the iSCSI CRC is not necessary.

[3.8](#). NAT traversal

iSCSI security utilizes transport mode ESP. As noted in [\[20\]](#), transport mode ESP cannot traverse NAT, even though ESP itself does not include IP header fields within its message integrity check. This is because the 16-bit TCP checksum is calculated based on a "pseudo-header" that includes IP header fields, and the checksum is protected by the IPsec

message integrity check. As a result, the TCP checksum will be invalidated by a NAT located between the iSCSI Initiator and Target.

Since TCP checksum calculation and verification is mandatory in both IPv4 and IPv6, it is not possible to omit checksum verification while remaining standards compliant. In order to enable traversal of NATs existing between iSCSI Initiators and Targets, while remaining in compliance, iSCSI/IPsec implementations MAY implement IPsec/IKE NAT traversal, as described in [20]-[23].

The IPsec/IKE NAT traversal specification [23] enables UDP encapsulation of IPsec to be negotiated 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. If this is done prior to calculating the IPsec message integrity check, TCP checksum verification will not fail.

[4.](#) Security considerations

IPsec IKE negotiation MUST negotiate an authentication method specified in the IKE [RFC 2409](#) [7]. In addition to IKE authentication, iSCSI implementations utilize their own authentication methods, such as those described in [48]. In this section, we discuss authentication issues.

[4.1.](#) IKE and iSCSI authentication

While iSCSI provides initial authentication, it does not provide per-packet authentication, integrity or replay protection. This implies that the identity verified in the iSCSI logon is not subsequently verified on reception of each packet.

With IPsec, when the identity asserted in IKE is authenticated, the resulting derived keys are used to provide per-packet authentication, integrity and replay protection. As a result, the identity verified in the IKE conversation is subsequently verified on reception of each packet.

Let us assume that the identity claimed in iSCSI logon is a user identity, while the identity claimed within IKE is a machine identity. Since only the machine identity is verified on a per-packet basis, there is no way for the recipient to verify that only the user authenticated via iSCSI logon is using the IPsec SA.

In fact, IPsec implementations that only support machine authentication typically have no way to distinguish between user traffic within the kernel. As a result, where machine authentication is used, once an iSCSI/IPsec SA is opened, another user on a multi-user machine may be

able to send traffic down the IPsec SA.

INTERNET-DRAFT

Securing iSCSI Using IPsec

18 August 2001

To limit the potential vulnerability, iSCSI security implementations MUST negotiate a separate IPsec Phase 2 SA for each iSCSI connection, with descriptors specific to that connection. This will prevent traffic for other iSCSI connections from travel within the IPsec SA negotiated for another iSCSI connection. As a result, if access to the TCP socket used for the iSCSI connection is exclusive to a single user, then access to the corresponding IPsec SA will also be exclusive, even if the IPsec implementation only supports machine authentication.

If the IPsec implementation supports user authentication, the user identity asserted within IKE will be verified on a per-packet basis, and stronger assurances can be provided. In this case, the user identity asserted within IKE will be verified on a per-packet basis. In order to provide segregation of traffic between users When user authentication is used, the sender MUST ensure that only traffic from that particular user is sent down the iSCSI SA. Enforcement of this restriction is the responsibility of the operating system.

[4.2.](#) Certificate authentication

When X.509 certificate authentication is chosen within IKE, the iSCSI Target is expected to use an IKE Certificate Request Payload (CRP) to request from the Initiator a certificate issued by a particular certificate authority or may use several CRPs if several certificate authorities are trusted and configured in its IPsec IKE authentication policy.

The iSCSI Target SHOULD be able to trust several certificate authorities in order to allow iSCSI Initiators to connect to it using their own certificate credential from their chosen PKI. Client and server side certificate revocation list checking MAY be enabled on a per-CA basis, since differences in revocation list checking exist between different PKI providers.

[4.3.](#) Machine versus user certificates

The certificate credentials provided by the iSCSI Initiator during the IKE negotiation MAY be those of the machine or of the iSCSI user. When machine authentication is used, the machine certificate is typically stored on the iSCSI Initiator and Target during an enrollment process.

When user certificates are used, the user certificate can be stored either on the machine or on a smartcard.

Since the value of a machine certificate is inversely proportional to the ease with which an attacker can obtain one under false pretenses, it is advisable that the machine certificate enrollment process be strictly controlled. For example, only administrators may have the ability to enroll a machine with a machine certificate.

While smartcard certificate storage lessens the probability of compromise of the private key, smartcards are not necessarily desirable in all situations. For example, some organizations deploying machine certificates use them so as to restrict use of non-approved hardware. Since user authentication can be provided within iSCSI logon (keeping in mind the weaknesses described earlier), support for machine authentication in IPsec makes it is possible to authenticate both the machine as well as the user.

In circumstances in which this dual assurance is considered valuable, enabling movement of the machine certificate from one machine to another, as would be possible if the machine certificate were stored on a smart card, may be undesirable.

Similarly, when user certificate are deployed, it is advisable for the user enrollment process to be strictly controlled. If for example, a user password can be readily used to obtain a certificate (either a temporary or a longer term one), then that certificate has no more security value than the password. To limit the ability of an attacker to obtain a user certificate from a stolen password, the enrollment period can be limited, after which password access will be turned off. Such a policy will prevent an attacker obtaining the password of an unused account from obtaining a user certificate once the enrollment period has expired.

[4.4.](#) Pre-shared keys

Use of pre-shared keys in IKE main mode is vulnerable to man-in-the-middle attacks when used with dynamically addressed Initiators. In main mode it is necessary for SKEYID_e to be used prior to the receipt of the identification payload. Therefore the selection of the pre-shared key may only be based on information contained in the IP header. However, where dynamic IP address assignment is typical, it is often not possible

to identify the required pre-shared key based on the IP address.

Thus when main mode pre-shared keys are used with iSCSI Targets whose address is dynamically assigned (such as desktop workstations), the same pre-shared key is shared by a group of Initiators and is no longer able to function as an effective shared secret. In this situation, neither the client nor the server identifies itself during IKE phase 1; it is only known that both parties are a member of the group with knowledge of the pre-shared key. This permits anyone with access to the group pre-shared key to act as a man-in-the-middle.

This vulnerability does not occur in aggressive mode since the identity payload is sent earlier in the exchange, and therefore the pre-shared key can be selected based on the identity. However, when aggressive mode is used the user identity is exposed and this is often considered

undesirable.

As a result, where main mode is used with pre-shared keys, unless iSCSI logon performs mutual authentication, the Target is not authenticated. This enables a rogue Target in possession of the group pre-shared key to successfully masquerade as the actual Target and mount a dictionary attack on legacy authentication methods such as CHAP [47]. Such an attack could potentially compromise many passwords at a time. This vulnerability is widely present in existing IPsec implementations.

To avoid this problem, iSCSI/IPsec implementations SHOULD NOT use a group pre-shared key for IKE authentication with main mode. If pre-shared keys are required, then aggressive mode SHOULD be used. IKE pre-shared authentication key values SHOULD be protected in a manner similar to the user's account password used in iSCSI logon.

5. References

- [1] Satran, J., et al., "iSCSI", Internet draft (work in progress), [draft-ietf-ips-iSCSI-06.txt](#), April 2001.
- [2] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [3] Kent, S., Atkinson, R., "IP Authentication Header", [RFC 2402](#), November 1998.

- [4] Kent, S., Atkinson, R., "IP Encapsulating Security Payload (ESP)", [RFC 2406](#), November 1998.
- [5] Piper, D., "The Internet IP Security Domain of Interpretation of ISAKMP", [RFC 2407](#), November 1998.
- [6] Atkinson, R., Kent, S., "Security Architecture for the Internet Protocol", [RFC 2401](#), November 1998.
- [7] Harkins, D., Carrel, D., "The Internet Key Exchange (IKE)", [RFC 2409](#), November 1998.
- [8] Rivest, R., "The MD5 Message-Digest Algorithm", [RFC 1321](#), April 1992.
- [9] Dobbertin, H., "The Status of MD5 After a Recent Attack", CryptoBytes Vol.2 No.2, Summer 1996.
- [10] Postel, J., "Transmission Control Protocol", STD 7, [RFC 793](#), September 1981.

- [11] Mogul, J., and S. Deering, "Path MTU Discovery", [RFC 1191](#), November 1990.
- [12] Knowles, S., "IESG Advice from Experience with Path MTU Discovery", [RFC 1435](#), March 1993.
- [13] McCann, J., Deering, S. and J. Mogul, "Path MTU Discovery for IP version 6", [RFC 1981](#), August 1996.
- [14] Lahey, K., "TCP Problems with Path MTU Discovery", [RFC 2923](#), September 2000.
- [15] Paxson, V., "End-to-end internet packet dynamics", IEEE Transactions on Networking 7,3 (June 1999) pg 277-292.
- [16] Stone J., Partridge, C., "When the CRC and TCP checksum disagree", ACM Sigcomm, Sept. 2000.
- [17] Cracking DES, O'Reilly & Associates, Sebastapol, CA 2000.

- [18] Krueger, M., et.al., "iSCSI Requirements and Design Considerations", [draft-ietf-ips-iscsi-reqmts-05.txt](#), Work in Progress, July 2001.
- [19] Krawczyk, H., Bellare, M. and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", [RFC 2104](#), February 1997.
- [20] Aboba, B., "IPsec-NAT Compatibility Requirements", [draft-ietf-IPsec-nat-reqts-00.txt](#), Work in Progress, June 2001.
- [21] Huttunen, A. et. al., "UDP Encapsulation of IPsec Packets", [draft-ietf-IPsec-udp-encaps-00.txt](#), June 2001
- [22] Dixon, W. et. al., "IPsec over NAT Justification for UDP Encapsulation", [draft-ietf-IPsec-udp-encaps-justification-00.txt](#), June 2001
- [23] Kivinen, T., et al., "Negotiation of NAT-Traversal in the IKE", Internet draft (work in progress), [draft-ietf-IPsec-nat-t-ike-00.txt](#), June 2001.
- [24] Madson, C., Glenn, R., "The Use of HMAC-MD5-96 within ESP and AH", [RFC 2403](#), November 1998.
- [25] Madson, C., Glenn, R., "The Use of HMAC-SHA-1-96 within ESP and AH", [RFC 2404](#), November 1998.

- [26] Rogaway, P., Black, J., "PMAC: Proposal to NIST for a parallelizable message authentication code", <http://csrc.nist.gov/encryption/modes/proposedmodes/pmac/pmac-spec.pdf>
- [27] Black, J., Halevi, S., Krawczyk, H., Krovetz, T., Rogaway, P., "UMAC: Fast and provably secure message authentication", Advances in Cryptology - CRYPTO '99, LNCS vol. 1666, pp. 216-233. Full version available from <http://www.cs.ucdavis.edu/~rogaway/umac>
- [28] "Descriptions of SHA-256, SHA-384, and SHA-512," <http://csrc.nist.gov/cryptval/shs/sha256-384-512.pdf>.

- [29] U.S. DoC/NIST, "Data encryption standard (DES)", FIPS 46-3, October 25, 1999.
- [30] U.S. DoC/NIST, "Guidelines for implementing and using the nbs data encryption standard", FIPS 74, Apr 1981.
- [31] Biham, E., Shamir, A., "Differential Cryptanalysis of DES- like cryptosystems", Journal of Cryptology Vol 4, Jan 1991.
- [32] Madson, C., Doraswamy, N., "The ESP DES-CBC Cipher Algorithm With Explicit IV", [RFC 2405](#), November 1998.
- [33] Pereira, R., Adams, R., "The ESP CBC-Mode Cipher Algorithms", [RFC 2451](#), November 1998.
- [34] Daemen, J., Rijman, V., "AES Proposal: Rijndael," NIST AES Proposal, June 1998. <http://csrc.nist.gov/encryption/aes/round2/AESAlgs/Rijndael/Rijndael.pdf>
- [35] Draft FIPS Publication ZZZZ, "Advanced Encryption Standard (AES)", U.S. DoC/NIST, summer 2001.
- [36] "Symmetric Key Block Cipher Modes of Operation," <http://www.nist.gov/modes>.
- [37] "Recommendation for Block Cipher Modes of Operation", National Institute of Standards and Technology (NIST) Special Publication 800-XX, CODEN: NSPUE2, U.S. Government Printing Office, Washington, DC, July 2001.
- [38] Frankel, S., Kelly, S., Glenn, R., "The AES Cipher Algorithm and Its Use with IPsec", Internet draft (work in progress), [draft-ietf-ipsec-ciph-aes-cbc-01.txt](#), May 2001.

- [39] Etienne, J., "The counter-mode and its use with ESP", Internet draft (work in progress), [draft-etienne-ipsec-esp-ctr-mode-00.txt](#), May 2001.
- [40] Lipmaa, H., Rogaway, P., Wagner, D., "CTR-MODE encryption", Comment

on mode of operations NIST, Jan 2001.

- [41] Schneier, B., J. Kelsey, D. Whiting, D. Wagner, C. Hall, and N. Ferguson, "Performance Comparison of the AES Submissions", <http://www.counterpane.com/AES-performance.html>
- [42] A. Bosselaers, "Performance of Pentium implementations", <http://www.esat.kuleuven.ac.be/~bosselae/>
- [43] Bellare, S., "An Issue With DES-CBC When Used Without Strong Integrity", Proceedings of the 32nd IETF, Danvers, MA, April 1995.
- [44] Krovetz, T., Black, J., Halevi, S., Hevia, A., Krawczyk, H., Rogaway, P., "UMAC: Message Authentication Code using Universal Hashing", Internet draft (work in progress), [draft-krovetz-umac-01.txt](#), October 2000.
- [45] Frankel, S., Kelly, S., "The Use of SHA-256, SHA-384, and SHA-512 within ESP, AH and IKE," Work in progress.
- [46] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0", [RFC 2246](#), November 1998.
- [47] Simpson, W., "PPP Challenge Handshake Authentication Protocol (CHAP)," [RFC 1994](#), August 1996.
- [48] Black, D., "iSCSI Security Requirements and SRP-based ESP keys", Internet draft (work in progress), [draft-black-ips-iscsi-security-00.txt](#), July 2001.
- [49] Steve Kent, IPsec sequence number extension proposal, IETF 50.
- [50] American National Standard for Financial Services X9.52-1998, "Triple Data Encryption Algorithm Modes of Operation", American Bankers Association, Washington, D.C., July 29, 1998.
- [51] Bellare, Desai, Jokipii, Rogaway, "A Concrete Treatment of Symmetric Encryption: Analysis of the DES Modes of Operation", 1997, <http://www-cse.ucsd.edu/users/mihir/>

Appendix A - Software Performance of IPsec Transforms

This Appendix provides data on the performance of IPsec encryption and authentication transforms in software. Since the performance of IPsec transforms is heavily implementation dependent, the data presented here may not be representative of performance in a given situation, and are presented solely for purposes of comparison.

[A.1](#) Authentication transforms

Table A-1 presents the cycles/byte required by the AES-PMAC, AES-CBC-MAC, AES-UMAC, HMAC-MD5, and HMAC-SHA1 algorithms at various packet sizes, implemented in software.

Data Size	AES- PMAC	AES-CBC- MAC	AES- UMAC	HMAC- MD5	HMAC- SHA1
64	31.22	26.02	19.51	93.66	109.27
128	33.82	28.62	11.06	57.43	65.04
192	34.69	26.02	8.67	45.09	48.56
256	33.82	27.32	7.15	41.63	41.63
320	33.3	27.06	6.24	36.42	37.46
384	33.82	26.88	5.42	34.69	34.69
448	33.45	26.76	5.39	32.71	31.96
512	33.82	26.67	4.88	31.22	30.57
576	33.53	26.59	4.77	30.64	29.48
640	33.3	26.54	4.42	29.66	28.62

INTERNET-DRAFT

Securing iSCSI Using IPsec

18 August 2001

Data Size	AES- PMAC	AES-CBC- MAC	AES- UMAC	HMAC- MD5	HMAC- SHA1
768	33.82	26.88	4.23	28.18	27.32
896	33.45	27.13	3.9	27.5	25.64
1024	33.5	26.67	3.82	26.99	24.71
1152	33.53	27.17	3.69	26.3	23.99
1280	33.56	26.8	3.58	26.28	23.67
1408	33.58	26.96	3.55	25.54	23.41
1500	33.52	26.86	3.5	25.09	22.87

Table A-1: Cycles/byte consumed by the AES-PMAC, AES-CBC-MAC, AES-UMAC, HMAC-MD5, and HMAC-SHA1 authentication algorithms at various packet sizes.

Source: Jesse Walker, Intel

(See also <http://www.cs.ucdavis.edu/~rogaway/umac/perf00.html> for additional data)

INTERNET-DRAFT

Securing iSCSI Using IPsec

18 August 2001

Table A-2 presents the cycles/second required by the AES-PMAC, AES-CBC-MAC, AES-UMAC, HMAC-MD5, and HMAC-SHA1 algorithms, implemented in software, assuming a 1500 byte packet.

Transform	Cycles/ octet (software)	Cycles/sec @ 100 Mbps	Cycles/sec @ 1 Gbps	Cycles/sec @ 10 Gbps
AES-UMAC (8 octets)	3.5	43,750,000	437,500,000	4.375 B
HMAC-SHA1 (20 octets)	22.87	285,875,000	2.8588 B	28.588 B
HMAC-MD5	25.09	313,625,000	3.1363 B	31.363 B
AES-CBC-MAC	26.86	335,750,000	3.358 B	33.575 B
AES-PMAC (8 octets)	33.52	419,000,000	4.19 B	41.900 B

Table A-2: Software performance of the HMAC-SHA1, HMAC-MD5, AES-CBC-MAC and AES-PMAC authentication algorithms at 100 Mbps, 1 Gbps, and 10 Gbps line rates (1500 byte packet).

Source: Jesse Walker, Intel

At speeds of 100 Mbps, AES-UMAC is implementable with only a modest processor, and the other algorithms are implementable, assuming that a single high-speed processor can be dedicated to the task. At 1 Gbps, only AES-UMAC is implementable on a single high-speed processor; multiple high speed processors (1+ Ghz) will be required for the other algorithms. At 10 Gbps, only AES-UMAC is implementable even with multiple high speed processors; the other algorithms will require a prodigious number of cycles/second. Thus at 10 Gbps, hardware acceleration will be required for all algorithms with the possible exception of AES-UMAC.

[A.2](#) Encryption and Authentication transforms

Table A-3 presents the cycles/byte required by the AES-CBC, AES-CTR and 3DES-CBC encryption algorithms (no MAC), implemented in software, for various packet sizes.

Data size	AES-CBC	AES-CTR	3DES-CBC
64	31.22	26.02	156.09
128	31.22	28.62	150.89
192	31.22	27.75	150.89
256	28.62	27.32	150.89
320	29.14	28.1	150.89
384	28.62	27.75	148.29
448	28.99	27.5	149.4
512	28.62	27.32	148.29
576	28.33	27.75	147.72

640	28.62	27.06	147.77
-----	-------	-------	--------

Data size	AES-CBC	AES-CTR	3DES-CBC
768	28.18	27.32	147.42
896	28.25	27.5	147.55
1024	27.97	27.32	148.29
1152	28.33	27.46	147.13
1280	28.1	27.58	146.99
1408	27.91	27.43	147.34
1500	27.97	27.53	147.85

Table A-3: Cycles/byte consumed by the AES-CBC, AES-CTR and 3DES-CBC

encryption algorithms at various packet sizes, implemented in software.

Source: Jesse Walker, Intel

Table A-4 presents the cycles/second required by the AES-CBC, AES-CTR and 3DES-CBC encryption algorithms (no MAC), implemented in software, at [100](#) Mbps, 1 Gbps, and 10 Gbps line rates (assuming a 1500 byte packet).

Transform	Cycles/ octet (software)	Cycles/sec @ 100 Mbps	Cycles/sec @ 1 Gbps	Cycles/sec @ 10 Gbps
AES-CBC	27.97	349,625,000	3.4963 B	34.963 B

AES-CTR	27.53	344,125,000	3.4413 B	34.413 B
3DES -CBC	147.85	1.84813 B	18.4813 B	184.813 B

Table A-4: Software performance of the AES-CBC, AES-CTR, and 3DES encryption algorithms at 100 Mbps, 1 Gbps, and 10 Gbps line rates (1500 byte packet).

Source: Jesse Walker, Intel

At speeds of 100 Mbps, AES-CBC and AES-CTR mode are implementable with a high-speed processor, while 3DES would require multiple high speed processors. At speeds of 1 Gbps, multiple high speed processors are required for AES-CBC and AES-CTR mode. At speeds of 1+ Gbps for 3DES, and 10 Gbps for all algorithms, implementation in software is infeasible, and hardware acceleration is required.

Table A-5 presents the cycles/byte required for combined encryption/authentication algorithms: AES CBC + CBCMAC, AES CTR + CBCMAC, AES CTR + UMAC, and AES-OCB at various packet sizes, implemented in software.

The diagram illustrates a 16-round Feistel network. The top horizontal line consists of 16 '+' signs, representing the rounds. Below this, the network is divided into four segments by vertical bars. The first segment contains the text 'AES' and is followed by a '+' sign. The second segment contains the text 'AES' and is followed by a '+' sign. The third segment contains the text 'AES' and is followed by a '+' sign. The fourth segment contains the text 'AES' and is followed by a '+' sign. The entire network is enclosed in a rectangular box.

Data size	CBC + CBCMAC	CTR + CBCMAC	CTR + UMAC	AES- OCB
64	119.67	52.03	52.03	57.23
128	70.24	57.23	39.02	44.23
192	58.97	55.5	36.42	41.63
256	57.23	55.93	35.12	40.32
320	57.23	55.15	33.3	38.5
384	57.23	55.5	32.95	37.29
448	58.72	55	32.71	37.17
512	58.54	55.28	32.52	36.42

Data size	AES CBC + CBCMAC	AES CTR + CBCMAC	AES CTR + UMAC	AES- OCB
576	57.81	55.5	31.8	37
640	57.75	55.15	31.74	36.42
768	57.67	55.5	31.65	35.99
896	57.61	55.75	31.22	35.68
1024	57.56	55.61	31.22	35.45
1152	57.52	55.21	31.22	35.55
1280	57.75	55.15	31.22	36.16
1408	57.47	55.34	30.75	35.24
1500	57.72	55.5	30.86	35.3

Table A-5: Cycles/byte of combined encryption/authentication algorithms: AES CBC + CBCMAC, AES CTR + CBCMAC, AES CTR + UMAC, and AES-OCB at various packet sizes, implemented in software.

INTERNET-DRAFT

Securing iSCSI Using IPsec

18 August 2001

Table A-6 presents the cycles/second required for the AES CBC + CBCMAC, AES CTR + CBCMAC, AES CTR + UMAC, and AES-OCB encryption and authentication algorithms operating at line rates of 100 Mbps, 1 Gbps and 10 Gbps, assuming 1500 byte packets.

Transform	Cycles/ octet (software)	Cycles/sec @ 100 Mbps	Cycles/sec @ 1 Gbps	Cycles/sec @ 10 Gbps
AES CBC + CBCMAC	57.72	721,500,000	7.215 B	72.15 B
AES CTR + CBCMAC	55.5	693,750,000	6.938 B	69.38 B
AES CTR + UMAC	30.86	385,750,000	3.858 B	38.58 B
AES-OCB	35.3	441,250,000	4.413 B	44.13 B

Table A-6: Cycles/second required for the AES CBC + CBCMAC, AES CTR + CBCMAC, AES CTR + UMAC, and AES-OCB encryption and authentication algorithms, operating at line rates of 100 Mbps, 1 Gbps and 10 Gbps, assuming 1500 octet packets.

Source: Jesse Walker, Intel

At speeds of 100 Mbps, the algorithms are implementable on a high speed processor. At speeds of 1 Gbps, multiple high speed processors are required, and none of the algorithms are implementable in software at 10 Gbps line rate.

Acknowledgments

Thanks to Steve Bellovin of AT&T Research for useful discussions of this problem space.

Aboba, et al.

Standards Track

[Page 31]

INTERNET-DRAFT

Securing iSCSI Using IPsec

18 August 2001

Authors' Addresses

Bernard Aboba
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

Phone: +1 425 936 6605
EMail: bernarda@microsoft.com

William Dixon
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

Phone: +1 425 703 8729
EMail: wdixon@microsoft.com

Joseph J. Tardo
Broadcom
[3151](#) Zanker Road
San Jose, CA 95134

Phone: +1 408 501 8445
Fax: +1 408 501 8460
EMail: jtardo@broadcom.com

Mark Bakke
Cisco Systems, Inc.
[6450](#) Wedgwood Road, Suite 130
Maple Grove, MN 55311

Phone: +1 763 398 1000
Fax: +1 763 398 1001
EMail: mbakke@cisco.com

Steve Senum
Cisco Systems, Inc.
[6450](#) Wedgwood Road, Suite 130
Maple Grove, MN 55311

Phone: <TBD>
Fax: +1 763 398 1001
EMail: ssenum@cisco.com

Howard Herbert
Intel Corporation
[5000](#) West Chandler Blvd.

Aboba, et al.

Standards Track

[Page 32]

INTERNET-DRAFT

Securing iSCSI Using IPsec

18 August 2001

M/S CH7-404
Chandler, Arizona 85226

Phone: +1 480 554 3116
EMail: howard.c.herbert@intel.com

Jesse Walker
Intel Corporation
[2211](#) NE 25th Avenue
Hillboro, Oregon 97124

Phone: +1 503 712 1849
Fax: +1 503 264 4843
Email: jesse.walker@intel.com

Julian Satran
IBM, Haifa Research Lab
MATAM - Advanced Technology Center
Haifa 31905, Israel

Phone +972 4 829 6264
EMail: Julian_Satran@vnet.ibm.com

Ofer Biran
IBM, Haifa Research Lab
MATAM - Advanced Technology Center
Haifa 31905, Israel

Phone +972 4 829 6253

Email: biran@il.ibm.com

Charles Kunzinger
IBM Corporation
Research Triangle Park, NC 27709

Phone: +1 919 254 4142
Email: kunzinge@us.ibm.com

David L. Black
EMC Corporation
[42](#) South Street
Hopkinton, MA 01748

Phone: +1 508 435 1000 x75140
Email: black_david@emc.com

Aboba, et al.

Standards Track

[Page 33]

INTERNET-DRAFT

Securing iSCSI Using IPsec

18 August 2001

Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in [BCP-11](#). Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

Full Copyright Statement

Copyright (C) The Internet Society (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 of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English. The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns. This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE 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 OR FITNESS FOR A PARTICULAR PURPOSE."

Aboba, et al.

Standards Track

[Page 34]

INTERNET-DRAFT

Securing iSCSI Using IPsec

18 August 2001

Expiration Date

This memo is filed as <[draft-aboba-ips-iscsi-security-00.txt](#)>, and expires February 19, 2002.

