IPS Working Group
INTERNET-DRAFT
Category: Standards Track
<draft-ietf-ips-security-14.txt>
22 July 2002

B. Aboba
Microsoft
Joshua Tseng
Nishan Systems
Jesse Walker
Intel
Venkat Rangan
Rhapsody Networks Inc.
Franco Travostino
Nortel Networks

Securing Block Storage Protocols over IP

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC 2026.

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

Abstract

This document discusses how to secure block storage and storage discovery protocols running over IP.

Table of Contents

1. Introduction

This specification discusses use of the IPsec protocol suite for protecting block storage protocols over IP networks (including iSCSI, iFCP and FCIP), as well as storage discovery protocols (iSNS and SLPv2).

1.1. iSCSI overview

iSCSI, described in [iSCSI], is a connection-oriented command/response protocol that runs over TCP, and is used to access disk, tape and other devices. iSCSI is a client-server protocol in which clients (Initiators) open connections to servers (Targets) and perform an iSCSI login.

This draft uses the SCSI terms Initiator and Target for clarity and to avoid the common assumption that clients have considerably less computational and memory resources than servers; the reverse is often the case for SCSI, as Targets are commonly dedicated devices of some form.

The iSCSI protocol has a text based negotiation mechanism as part of its initial (login) procedure. The mechanism is extensible in what can be negotiated (new text keys and values can be defined) and also in the number of negotiation rounds (e.g., to accommodate functionality such as challenge-response authentication). While the iSCSI login 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.

iSCSI Initiators and Targets are layer 5 entities that are independent of TCP ports and IP addresses. Initiators and Targets have names whose syntax is defined in [iSCSIName]. iSCSI sessions between a given Initiator and Target are run over one or more TCP connections between those entities. That is, the login process establishes an association between an iSCSI Session and the TCP connection(s) over which iSCSI PDUs will be carried.

1.2. iFCP overview

iFCP, defined in [iFCP], is a gateway-to-gateway protocol, which provides transport services to Fibre Channel devices over a TCP/IP network. iFCP allows interconnection and networking of existing Fibre Channel devices at wire speeds over an IP network. iFCP implementations emulate fabric services in order to improve fault tolerance and scalability by fully leveraging IP technology. Each TCP connection is used to support storage traffic between a unique pair of Fibre Channel N_PORTs.

iFCP does not have a native, in-band security mechanism. Rather, it relies upon the IPsec protocol suite to provide data confidentiality and authentication services, and IKE as the key management protocol. iFCP uses TCP to provide congestion control, error detection and error recovery.

1.3. FCIP overview

FCIP, defined in [FCIP], is a pure FC encapsulation protocol that transports FC frames. Current specification work intends this for interconnection of Fibre Channel switches over TCP/IP networks, but the protocol is not inherently limited to connecting FC switches. FCIP differs from iFCP in that no interception or emulation of fabric services is involved. One or more TCP connections are bound to an FCIP Link, which is used to realize Inter-Switch Links (ISLs) between pairs of Fibre Channel entities.

FCIP does not have a native, in-band security mechanism. Rather, it relies upon the IPsec protocol suite to provide data confidentiality and authentication services, and IKE as the key management protocol. FCIP uses TCP to provide congestion control, error detection and error recovery.

1.4. IPsec overview

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 [RFC2401], IKE [RFC2409], IPsec Authentication Header (AH) [RFC2402] and IPsec Encapsulating Security Payload (ESP) [RFC2406] documents. IKE is the key management protocol while AH and ESP are used to protect IP traffic.

An IPsec SA is a one-way security association, uniquely identified by the 3-tuple: Security Parameter Index (SPI), protocol (ESP) and destination IP. The parameters for an IPsec security association are typically established by a key management protocol. These include the encapsulation mode, encapsulation type, session keys and SPI values.

IKE is a two phase negotiation protocol based on the modular exchange of messages defined by ISAKMP [RFC2408],and the IP Security Domain of Interpretation (DOI) [RFC2407]. IKE has two phases, and accomplishes the following functions:

[1] Protected cipher suite and options negotiation - using keyed MACs and encryption and anti-replay mechanisms

[2] Master key generation - such as via MODP Diffie-Hellman calculations

[3] Authentication of end-points

[4] IPsec SA management (selector negotiation, options negotiation, create, delete, and rekeying)

Items 1 through 3 are accomplished in IKE Phase 1, while item 4 is handled in IKE Phase 2.

An IKE Phase 2 negotiation is performed to establish both an inbound and an outbound IPsec SA. The traffic to be protected by an IPsec SA is determined by a selector which has been proposed by the IKE Initiator and accepted by the IKE Responder. In IPsec transport mode, the IPsec SA selector can be a "filter" or traffic classifier, defined as the 5-tuple: <Source IP address, Destination IP address, transport protocol (UDP/SCTP/TCP), Source port, Destination port>. The successful establishment of a IKE Phase-2 SA results in the creation of two uni-directional IPsec SAs fully qualified by the tuple <Protocol (ESP/AH), destination address, SPI>.

The session keys for each IPsec SA are derived from a master key, typically via a MODP Diffie-Hellman computation. Rekeying of an existing IPsec SA pair is accomplished by creating two new IPsec SAs, making them active, and then optionally deleting the older IPsec SA pair. Typically the new outbound SA is used immediately, and the old inbound SA is left active to receive packets for some locally defined time, perhaps 30 seconds or 1 minute.

1.5. Terminology

Fibre Channel

Fibre Channel (FC) is a gigabit speed networking technology primarily used to implement Storage Area Networks (SANs), although it also may be used to transport other frames types as well, including IP. FC is standardized under American National Standard for Information Systems of the InterNational Committee for Informational Technology Standards (ANSI-INCITS) in its T11 technical committee.

FCIP Fibre Channel over IP (FCIP) is a protocol for interconnecting Fibre Channel islands over IP Networks so as to form a unified SAN in a single Fibre Channel fabric. The principal FCIP interface point to the IP Network is the FCIP Entity. The FCIP Link represents one or more TCP connections that exist between a pair of FCIP Entities.

HBA Host Bus Adapter (HBA) is a generic term for a SCSI interface to other device(s); it's roughly analogous to the term Network Interface Card (NIC) for a TCP/IP network interface, except that HBAs generally have on-board SCSI implementations, whereas most NICs do not implement TCP, UDP, or IP.

iFCP iFCP is a gateway-to-gateway protocol, which provides Fibre Channel fabric services to Fibre Channel devices over a TCP/IP network.

IP block storage protocol
Where used within this document, the term "IP block storage protocol" applies to all block storage protocols running over IP, including iSCSI, iFCP and FCIP.

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

iSNS The Internet Storage Name Server (iSNS) protocol provides for discovery and management of iSCSI and Fibre Channel (FCP) storage devices. iSNS applications store iSCSI and FC device attributes and monitor their availability and reachability, providing a consolidated information repository for an integrated IP block storage network. iFCP requires iSNS for discovery and management, while iSCSI may use iSNS for discovery, and FCIP does not use iSNS.

Initiator The iSCSI Initiator connects to the Target on well-known TCP port 3260. 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.

1.6. Requirements language

In this document, the key words "MAY", "MUST, "MUST NOT", "OPTIONAL", "RECOMMENDED", "SHALL", "SHALL NOT", "SHOULD", and "SHOULD NOT", are to be interpreted as described in [RFC2119].

Note that requirements specified in this document apply only to use of IPsec and IKE with IP block storage protocols. Thus, these requirements do not apply to IPsec implementations in general. Implementation requirements language should therefore be assumed to relate to the availability of features for use with IP block storage security only.

Although the security requirements in this document are already incorporated into the iSCSI [iSCSI], iFCP [iFCP] and FCIP [FCIP] standards track documents, they are reproduced here for convenience. In the event of a discrepancy, the individual protocol standards track documents take precedence.

2. Block storage protocol security

2.1. Security requirements

IP Block storage protocols such as iSCSI, iFCP and FCIP are used to transmit SCSI commands over IP networks. Therefore, both the control and data packets of these IP block storage protocols are vulnerable to attack. Examples of attacks include:

[1] An adversary may attempt to acquire confidential data and identities by snooping data packets.

[2] An adversary may attempt to modify packets containing data and control messages.

[3] An adversary may attempt to inject packets into an IP block storage connection.

[4] An adversary may attempt to hijack TCP connection(s) corresponding to an IP block storage session.

[5] An adversary may launch denial of service attacks against IP block storage devices such as by sending a TCP reset.

[6] An adversary may attempt to disrupt security negotiation process, in order to weaken the authentication, or gain access to user passwords. This includes disruption of application-layer authentication negotiations such as iSCSI Login.

[7] An adversary may attempt to impersonate a legitimate IP block storage entity.

[8] An adversary may launch a variety of attacks (packet modification or injection, denial of service) against the discovery (SLPv2, [RFC2608]) or discovery and management (iSNS, [iSNS]) process. iSCSI can use SLPv2 or iSNS. FCIP only uses SLPv2, and iFCP only

uses iSNS.

Since iFCP and FCIP devices are the last line of defense for a whole Fibre Channel island, the above attacks, if successful, could compromise the security of all the Fibre Channel hosts behind the devices.

To address the above threats, IP block storage security protocols must support confidentiality, data origin authentication, integrity, and replay protection on a per-packet basis. Confidentiality services are important since IP block storage traffic may traverse insecure public networks. The IP block storage security protocols must support perfect forward secrecy in the rekeying process.

Bi-directional authentication of the communication endpoints MUST be provided. There is no requirement that the identities used in authentication be kept confidential (e.g., from a passive eavesdropper).

For a security protocol to be useful, CPU overhead and hardware availability must not preclude implementation at 1 Gbps today. Implementation feasibility at 10 Gbps is highly desirable, but may not be demonstrable at this time. These performance levels apply to aggregate throughput, and include all TCP connections used between IP block storage endpoints. IP block storage communications typically involve multiple TCP connections. Performance issues are discussed further in Appendix B.

Enterprise data center networks are considered mission-critical facilities that must be isolated and protected from possible security threats. Such networks are often protected by security gateways, which at a minimum provide a shield against denial of service attacks. The IP block storage security architecture should be able to leverage the protective services of the existing security infrastructure, including firewall protection, NAT and NAPT services, and VPN services available on existing security gateways.

When iFCP or FCIP devices are deployed within enterprise networks, IP addresses will be typically be statically assigned as is the case with most routers and switches. Consequently, support for dynamic IP address assignment, as described in [DHCPIPsec], will typically not be required, although it cannot be ruled out. Such facilities will also be relevant to iSCSI hosts whose addresses are dynamically assigned. As a result, the IP block storage security protocols MUST NOT introduce additional security vulnerabilities where dynamic address assignment is supported.

While IP block storage security is mandatory to implement, it is not mandatory to use. The security services used depend on the configuration and security policies put in place. For example,

configuration will influence the authentication algorithm negotiated within iSCSI Login, as well as the security services (confidentiality, data origin authentication, integrity, replay protection) and transforms negotiated when IPsec is used to protect IP block storage protocols such as iSCSI, iFCP and FCIP.

FCIP implementations may allow enabling and disabling security mechanisms at the granularity of an FCIP Link. For iFCP, the granularity corresponds to an iFCP Portal. For iSCSI, the granularity of control is typically that of an iSCSI session, although it is possible to exert control down to the granularity of the destination IP address and TCP port.

Note that with IPsec, security services are negotiated at the granularity of an IPsec SA, so that IP block storage connections requiring a set of security services different from those negotiated with existing IPsec SAs will need to negotiate a new IPsec SA. Separate IPsec SAs are also advisable where quality of service considerations dictate different handling of IP block storage connections. Attempting to apply different quality of service to connections handled by the same IPsec SA can result in reordering, and falling outside the replay window. For a discussion of the issues, see [RFC2983].

IP block storage protocols can be expected to carry sensitive data and provide access to systems and data that require protection against security threats. SCSI and Fibre Channel currently contain little in the way of security mechanisms, and rely on physical security, administrative security, and correct configuration of the communication medium and systems/devices attached to it for their security properties.

For most IP networks, it is inappropriate to assume physical security, administrative security, and correct configuration of the network and all attached nodes (a physically isolated network in a test lab may be an exception). Therefore, authentication SHOULD be used by IP block storage protocols (e.g., iSCSI SHOULD use one of its inband authentication mechanisms or the authentication provided by IKE) in order to provide a minimal assurance that connections have initially been opened with the intended counterpart.

iSNS, described in [iSNS], is required in all iFCP deployments. iSCSI may use iSNS for discovery, and FCIP does not use iSNS. iSNS applications store iSCSI and FC device attributes and monitor their availability and reachability, providing a consolidated information repository for an integrated IP block storage network. The iSNS specification defines mechanisms to secure communication between an iSNS server and its clients.

2.2. Resource constraints

iFCP and FCIP devices will typically be embedded systems deployed on racks in air-conditioned data center facilities. Such embedded systems may include hardware chipsets to provide data encryption, authentication, and integrity processing. Therefore, memory and CPU resources are generally not a constraining factor.

iSCSI will be implemented on a variety of systems ranging from large servers running general purpose operating systems to embedded host bus adapters (HBAs). In general, a host bus adapter is the most constrained iSCSI implementation environment, although an HBA may draw upon the resources of the system to which it is attached in some cases (e.g., authentication computations required for connection setup). More resources should be available to iSCSI implementations for embedded and general purpose operating systems. The following guidelines indicate the approximate level of resources that authentication, keying, and rekeying functionality can reasonably expect to draw upon:

- Low power processors with small word size are generally not used, as power is usually not a constraining factor, with the possible exception of HBAs, which can draw upon the computational resources of the system into which they are inserted). Computational horsepower should be available to perform a reasonable amount of exponentiation as part of authentication and key derivation for connection setup. The same is true of rekeying, although the ability to avoid exponentiation for rekeying may be desirable (but is not an absolute requirement).

- RAM and/or flash resources tend to be constrained in embedded implementations. 8-10 MB of code and data for authentication, keying, and rekeying is clearly excessive, 800-1000 KB is clearly larger than desirable, but tolerable if there is no other alternative and 80-100 KB should be acceptable. These sizes are intended as rough order of magnitude guidance, and should not be taken as hard targets or limits (e.g., smaller code sizes are always better). Software implementations for general purpose operating systems may have more leeway.

The primary resource concern for implementation of authentication and keying mechanisms is code size, as iSCSI assumes that the computational horsepower to do exponentiations will be available.

There is no dominant iSCSI usage scenario - the scenarios range from a single connection constrained only by media bandwidth to hundreds of Initiator connections to a single Target or communication endpoint. SCSI sessions and hence the connections they use tend to be relatively long lived; for disk storage, a host typically opens a SCSI connection

on boot and closes it on shutdown. Tape session length tends to be measured in hours or fractions thereof (i.e., rapid fire sharing of the same tape device among different Initiators is unusual), although tape robot control sessions can be short when the robot is shared among tape drives. On the other hand, tape will not see a large number of Initiator connections to a single Target or communication endpoint, as each tape drive is dedicated to a single use at a single time, and a dozen tape drives is a large tape device.

2.3. Security protocol

2.3.1. Transforms

All IP block storage security compliant implementations MUST support IPsec ESP [RFC2406] to provide security for both control packets and data packets, as well as the replay protection mechanisms of IPsec. When ESP is utilized, per-packet data origin authentication, integrity and replay protection MUST be used.

To provide confidentiality with ESP, ESP with 3DES in CBC mode [RFC2451] MUST be supported, and AES in Counter mode, as described in [AESCTR], SHOULD be supported. To provide data origin authentication and integrity with ESP, HMAC-SHA1 [RFC2404] MUST be supported, and AES in CBC MAC mode with XCBC extensions [AESXCBC] SHOULD be supported. DES in CBC mode SHOULD NOT be used due to its inherent weakness. ESP with NULL encryption MUST be supported for authentication.

2.3.2. IPsec modes

Conformant IP block storage protocol implementations MUST support ESP [RFC2406] in tunnel mode and MAY implement IPsec with ESP in transport mode.

2.3.3. IKE

Conformant IP block storage security implementations MUST support IKE [RFC2409] for peer authentication, negotiation of security associations, and key management, using the IPsec DOI [RFC2407]. Manual keying MUST NOT be used since it does not provide the necessary rekeying support. Conformant IP block storage security implementations MUST support peer authentication using a pre-shared key, and MAY support certificate-based peer authentication using digital signatures. Peer authentication using the public key encryption methods outlined in IKE's sections 5.2 and 5.3 [RFC2409] SHOULD NOT be used.

Conformant IP block storage security implementations MUST support IKE Main Mode and SHOULD support Aggressive Mode. IKE Main Mode with pre-shared key authentication SHOULD NOT be used when either of the peers

use a dynamically assigned IP address. While Main Mode with pre-shared key authentication offers good security in many cases, situations where dynamically assigned addresses are used force use of a group pre-shared key, which is vulnerable to man-in-the-middle attack.

When digital signatures are used for authentication, either IKE Main Mode or IKE Aggressive Mode MAY be used. In all cases, access to locally stored secret information (pre-shared key, or private key for digital signing) must be suitably restricted, since compromise of the secret information nullifies the security properties of the IKE/IPsec protocols.

When digital signatures are used to achieve authentication, an IKE negotiator SHOULD use IKE Certificate Request Payload(s) to specify the certificate authority (or authorities) that are trusted in accordance with its local policy. IKE negotiators SHOULD check the pertinent Certificate Revocation List (CRL) before accepting a PKI certificate for use in IKE's authentication procedures.

The IPsec DOI [RFC2407] provides for several types of identification data. Within IKE Phase 1, for use within the IDii and IDir payloads, conformant IP block storage security implementations MUST support the ID_IPV4_ADDR, ID_IPV6_ADDR (if the protocol stack supports IPv6) and ID_FQDN Identity Payloads. iSCSI security implementations SHOULD support the ID_USER_FQDN Identity Payload; other IP block storage protocols (iFCP, FCIP) SHOULD NOT use the ID_USER_FQDN Identity Payload. Identities other than ID_IPV4_ADDR and ID_IPV6_ADDR (such as ID_FQDN or ID_USER_FQDN) SHOULD be employed in situations where Aggressive mode is utilized along with pre-shared keys and IP addresses are dynamically assigned. The IP Subnet, IP Address Range, ID_DER_ASN1_DN, ID_DER_ASN1_GN formats SHOULD NOT be used for IP block storage protocol security; The ID_KEY_ID Identity Payload MUST NOT be used. As described in [RFC2407], within Phase 1 the ID port and protocol fields MUST be set to zero or to UDP port 500. Also, as noted in [RFC2407]:

> When an IKE exchange is authenticated using certificates (of any format), any ID's 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 used by IP block storage protocol implementations 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, ID_IPV6_ADDR) and SHOULD NOT use the IP Subnet or IP Address Range formats. Other ID payload formats MUST NOT be used.

Since IPsec acceleration hardware may only be able to handle a limited number of active IKE Phase 2 SAs, Phase 2 delete messages may be sent

for idle SAs, as a means of keeping the number of active Phase 2 SAs to a minimum. The receipt of an IKE Phase 2 delete message SHOULD NOT be interpreted as a reason for tearing down an IP block storage connection. Rather, it is preferable to leave the connection up, and if additional traffic is sent on it, to bring up another IKE Phase 2 SA to protect it. This avoids the potential for continually bringing connections up and down.

2.3.4. Security policy configuration

One of the goals of this specification is to enable a high level of interoperability without requiring extensive configuration. This section provides guidelines on setting of IKE parameters so as to enhance the probability of a successful negotiation. It also describes how information on security policy configuration can be provided so as to further enhance the chances of success.

To enhance the prospects for interoperability, some of the actions to consider include:

[1] Transform restriction. Since support for 3DES-CBC and HMAC-SHA1 is required of all implementations, offering these transforms enhances the probability of a successful negotiation. If AES-CTR [AESCTR] with XCBC-MAC [AESXCBC] is supported, this transform combination will typically be preferred, with 3DES-CBC/HMAC-SHA1 as a secondary offer.

[2] Group Restriction. If 3DES-CBC/HMAC-SHA1 is offered, and DH groups are offered, then it is recommended that a DH group of at least 1024 bits be offered along with it. If AES-CTR/XCBC-MAC is the preferred offer, and DH groups are offered, then it is recommended that a DH group of at least 2048 bits be offered along with it, as noted in [KeyLen]. If perfect forward secrecy is required in Quick Mode, then it is recommended that the QM PFS DH group be the same as the IKE Phase 1 DH group. This reduces the total number of combinations, enhancing the chances for interoperability.

[3] Key lifetimes. If a key lifetime is offered that is longer than desired, then rather than causing the IKE negotiation to fail, it is recommended that the Responder consider the offered lifetime as a maximum, and accept it. The key can then use a lesser value for the lifetime, and utilize a Lifetime Notify in order to inform the other peer of lifetime expiration.

Even when the above advice is taken, it still may be useful to be able to provide additional configuration information in order to enhance the chances of success, and it is useful to be able to manage security

configuration regardless of the scale of the deployment.

For example, it may be desirable to configure the security policy of an IP block storage device. This can be done manually or automatically via a security policy distribution mechanism. Alternatively, it can be supplied via iSNS or SLPv2. If an IP block storage endpoint can obtain the required security policy by other means (manually, or automatically via a security policy distribution mechanism) then it need not request this information via iSNS or SLPv2. However, if the required security policy configuration is not available via other mechanisms, iSNS or SLPv2 can be used to obtain it.

It may also be helpful to obtain information about the peer configuration. While it is generally possible to negotiate security parameters within IKE, there are situations in which incompatible parameters can cause the IKE negotiation to fail. In this case, it can be helpful to obtain information about the preferences of the peer prior to initiating IKE. This information can be provided via SLPv2 or iSNS. Information in this category includes:

[4] IPsec or cleartext support. The minimum piece of peer configuration required is whether an IP block storage endpoint requires IPsec or cleartext. This cannot be determined from the IKE negotiation alone without risking a long timeout, which is highly undesirable for a disk access protocol.

[5] Perfect Forward Secrecy (PFS) support. It is helpful to know whether a peer allows PFS, since an IKE Phase 2 Quick Mode can fail if an Initiator proposes PFS to a Responder that does not allow it.

[6] Preference for tunnel mode. While it is legal to propose both transport and tunnel mode within the same offer, not all IKE implementations will support this. As a result, it is useful to know whether a peer prefers tunnel mode or transport mode, so that it is possible to negotiate the preferred mode on the first try.

[7] Main Mode and Aggressive Mode support. Since the IKE negotiation can fail if a mode is proposed to a peer that doesn't allow it, it is helpful to know which modes a peer allows, so that an allowed mode can be negotiated on the first try.

Since iSNS or SLPv2 can be used to distribute IPsec security policy and configuration information for use with IP block storage protocols, these discovery protocols would constitute a 'weak link' were they not secured at least as well as the protocols whose security they configure. Since the major vulnerability is packet modification and replay, when iSNS or SLPv2 are 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 discovery protocol.

2.4. iSCSI authentication

2.4.1. CHAP

Compliant iSCSI implementations MUST implement the CHAP authentication method [RFC1994] (see [iSCSI], section 10.5), which includes support for bi-directional authentication.

When CHAP is performed over non-encrypted channel, it is vulnerable to an off-line dictionary attack. Implementations MUST support random CHAP secrets of up to 128 bits, including the means to generate such secrets and to accept them from an external generation source. Implementations MUST NOT provide secret generation (or expansion) means other than random generation.

If CHAP is used with secret smaller than 96 bits, then IPsec encryption (according to the implementation requirements in [iSCSI] section 7.3.2) MUST be used to protect the connection. Moreover, in this case IKE authentication with group pre-shared keys MUST NOT be used. When CHAP is used with a secret smaller then 96 bits, a compliant implementation MUST NOT continue with the iSCSI login unless it can verify that IPsec encryption is being used to protect the connection.

Initiators MUST NOT reuse the CHAP challenge sent by the Responder for the other direction of a bi-directional authentication. Responders MUST check for this condition and close the iSCSI TCP connection if it occurs.

A Responder MUST NOT send its CHAP response if the Initiator has not successfully authenticated. For example, the following exchange:

```
I->R     CHAP_A(A1,A2,...)
R->I     CHAP_A, CHAP_C, CHAP_I
I->R     CHAP_A, CHAP_C, CHAP_I
```

MUST result in the Responder (Target) closing the iSCSI TCP connection because the Initiator has failed to authenticate (there is no CHAP_R in the third message).

These requirements prevent reflection attacks in which the Initiator uses the same CHAP challenge as the Target and reflects the Target's response back to the Target, thereby authenticating the Initiator without requiring the Initiator to know the CHAP secret.

Note that RADIUS [RFC2865] does not support bi-directional CHAP authentication. Therefore, while a Target acting as a RADIUS client will be able to verify the Initiator Response, it will not be able to respond to an Initiator challenge unless it has access to the shared secret by some other means.

2.4.2. SRP

iSCSI implementations MAY implement the SRP authentication method [RFC2945]. The strength of SRP security is dependent on the characteristics of the group being used (i.e., the prime modulus N and generator g). As described in [RFC2945], N is required to be a Sophie-German prime (of the form N = 2q + 1, where q is also prime) the generator g is a primitive root of GF(n) [SRPNDSS]. For use in iSCSI authentication, the prime modulus N MUST be at least 768 bits.

Upon receiving N and g from the Target, the Initiator MUST verify that they satisfy the above requirements (and abort the connection otherwise). This verification MAY start by trying to match them with a well-known group that satisfies the above requirements. SRP well-known groups are included in Appendix A.

2.5. SLPv2 Security

Both iSCSI and FCIP protocols use SLPv2 as a way to discover peer entities and management servers. SLPv2 may also be used to provide information on peer security configuration. When SLPv2 is deployed, the SA advertisements as well as UA requests and/or responses are subject to the following security threats:

[1] An attacker could insert or alter SA advertisements or a response to a UA request in order to masquerade as the real peer or launch a denial of service attack.

[2] An attacker could gain knowledge about an SA or a UA through snooping, and launch an attack against the peer. Given the potential value of iSCSI targets and FCIP entities, leaking of such information not only increases the possibility of an attack over the network; there is also the risk of physical theft.

[3] An attacker could spoof a DAAdvert. This could cause UAs and SAs to use a rogue DAs.

To address these threats, the following capabilities are required:

[a] Service information, as included in SrvRply, AttrRply, SrvReg and SrvDereg messages, needs to be kept confidential.

[b] The UA has to be able to distinguish between legitimate and illegitimate service information from SrvRply and AttrRply messages. In the SLPv2 security model SAs are trusted to sign data.

[c] The DA has to be able to distinguish between legitimate and illegitimate SrvReg and SrvDereg messages.

[d] The UA has to be able to distinguish between legitimate and illegitimate DA Advertisements. This allows the UA to avoid rogue DAs which will return incorrect data or no data at all. In the SLPv2 security model, UAs trust DAs to store, answer queries on and forward data on services, but not necessarily to originate it.

[e] SAs may have to trust DAs, especially if 'mesh-enhanced' SLPv2 is used. In this case, SAs register with only one DA and trust that this DA will forward the registration to others.

By itself, SLPv2 security, defined in [RFC2608], does not satisfy these security requirements. SLPv2 only provides end-to-end authentication, but does not support confidentiality. In SLPv2 authentication there is no way to authenticate 'zero result responses'. This enables an attacker to mount a denial of service attack by sending UAs a 'zero results' SrvRply or AttrRply as if from a DA with whose source address corresponds to a legitimate DAAdvert.

In all cases, there is a potential for denial of service attack against protocol service providers, but such an attack is possible even in the absence of SLPv2 based discovery mechanisms.

2.5.1. SLPv2 security protocol

SLPv2 message types include: SrvRqst, SrvRply, SrvReg, SrvDereg, SrvAck, AttrRqst, AttrRply, DAAdvert, SrvTypeRqst, SrvTypeRply, SAAdvert. SLPv2 requires that UAs and SAs support SrvRqst, SrvRply, and DAAdvert. SAs must additionally support SrvReg, SrvAck, and SAAdvert.

Where no DA exists, the SrvRqst is multicast, but the SrvRply is sent via unicast UDP. DAAdverts are also multicast. However, all other SLPv2 messages are sent via UDP unicast.

In order to provide the required security functionality, iSCSI and FCIP security implementations SHOULD protect SLPv2 messages sent via unicast using IPsec ESP with a non-null transform. SLPv2 authentication blocks (carrying digital signatures), described in [RFC2608] MAY also be used to authenticate unicast and multicast messages.

The usage of SLPv2 by iSCSI is described in [iSCSISLP]. iSCSI Initiators and Targets may enable IKE mechanisms to establish identity. In addition, a subsequent user-level iSCSI session login can protect the Initiator-Target nexus. This will protect them from any compromise of security in the SLPv2 discovery process.

The usage of SLPv2 by FCIP is described in [FCIPSLP]. FCIP Entities assume that once the IKE identity of a peer is established, the FCIP Entity Name carried in FCIP Short Frame is also implicitly accepted as the authenticated peer. Any such association between the IKE identity and the FCIP Entity Name is administratively established.

For use in securing SLPv2, when digital signatures are used to achieve authentication in IKE, an IKE negotiator SHOULD use IKE Certificate Request Payload(s) to specify the certificate authority (or authorities) that are trusted in accordance with its local policy. IKE negotiators SHOULD check the pertinent Certificate Revocation List (CRL) before accepting a PKI certificate for use in IKE's authentication procedures. If key management of SLPv2 DAs needs to be coordinated with the SAs and the UAs as well as the protocol service implementations, one may use certificate based key management, with a shared root CA.

One of the reasons for utilizing IPsec for SLPv2 security is that is more likely that certificates will be deployed for IPsec than for SLPv2. This both simplifies SLPv2 security and makes it more likely that it will be implemented interoperably and more importantly, that it will be employed. As a result, it is desirable that little additional effort be required to enable IPsec protection of SLPv2.

However, just because a certificate is trusted for use with IPsec does not necessarily imply that the host is authorized to perform SLPv2 operations. When using IPsec to secure SLPv2, it may be desirable to distinguish between certificates appropriate for use by UAs, SAs, and DAs. For example, while a UA might be allowed to use any certificate conforming to IKE certificate policy, the certificate used by an SA might indicate that it is a legitimate source of service advertisements. Similarly, a DA certificate might indicate that it is a valid DA. This can be accomplished by using special CAs to issue certificates valid for use by SAs and DAs; alternatively SA and DA authorizations can be employed.

Assume that the policy for issuing and distributing SLPv2 authorized certificates to SAs and DAs limits them only to legitimate SAs and DAs. In this case, IPsec is used to provide SLPv2 security as follows:

[f] SLPv2 messages sent via unicast are IPsec protected, using ESP with a non-null transform.

[g] SrvRply and AttrRply messages from either a DA or SA are unicast to UAs. Assuming that the SA used a certificate authorized for SLPv2 service advertisement in establishing the IKE Phase 1 SA, or that the DA used a certificate authorized for DA usage, the UA can accept the information sent, even if it has no SLPv2 authentication block.

[h] SrvReg and SrvDereg messages from a SA are unicast to DAs. Assuming that the SA used a certificate authorized for SLPv2 service advertisement in establishing the IKE Phase 1 SA, the DA can accept the de/registration even if it has no SLPv2 authentication block. Typically, the SA will check the DA authorization prior to sending the service advertisement.

[i] Multicast DAAdverts can be considered advisory. The UA will attempt to contact DAs via unicast. Assuming that the DA used a certificate authorized for SLPv2 DAAdverts in establishing the IKE Phase 1 SA, the UA can accept the DAAdvert even if it has no SLPv2 authentication block.

[j] SAs can accept DAAdverts as described in i.

2.5.2. Confidentiality of service information

Since SLPv2 messages can contain information that can potentially reveal the vendor of the device or its other associated characteristics, revealing service information constitutes a security risk. As an example, the FCIP Entity Name may reveal a WWN from which an attacker can learn potentially useful information about the Entity's characteristics.

The SLPv2 security model assumes that service information is public, and therefore does not provide for confidentiality. However, storage devices represent mission critical infrastructure of substantial value, and so iSCSI and FCIP security implementations MUST support confidentiality as well as authentication of unicast SLPv2 messages.

Assuming that all unicast SLPv2 messages are protected by IPsec, and that confidentiality is provided, then the risk of disclosure can be limited to SLPv2 messages sent via multicast, namely the SrvRqst and DAAdvert.

The information leaked in a multicast SrvRqst depends on the level of detail in the query. If leakage is a concern, then a DA can be provided. If this is not feasible, then a general query can be sent via multicast, and then further detail can be obtained from the replying entities via additional unicast queries, protected by IPsec.

Information leakage via a multicast DAAdvert is less of a concern than the authenticity of the message, since knowing that a DA is present on the network only enables an attacker to know that SLPv2 is in use, and possibly that a directory service is also present. This information is not considered very valuable.

2.5.3. SLPv2 security implications

Through the definition of security attributes, it is possible to use SLPv2 to distribute information about security settings for IP block storage entities. SLPv2 distribution of security policy is not necessary if the security settings can be determined by other means, such as manual configuration or IPsec security policy distribution. If an entity has already obtained its security configuration via other mechanisms, then it MUST NOT request security policy via SLPv2.

Where SLPv2 is used to provide security policy information for use with IP block storage protocols, SLPv2 MUST be protected by IPsec as described in this document. Where SLPv2 is not used to distribute security policy information, implementations MAY implement SLPv2 security as described in this document.

Where SLPv2 is used, but security is not implemented, IP block storage protocol implementations MUST support a negative cache for authentication failures. This allows implementations to avoid continually contacting discovered endpoints which fail authentication within IPsec or at the application layer (in the case of iSCSI Login). The negative cache need not be maintained within the IPsec implementation, but rather within the IP block storage protocol implementation.

Since this document proposes that hop-by-hop security be used as the primary mechanism to protect SLPv2, UAs have to trust DAs to accurately relay data from SAs. This is a change to the SLPv2 security model described in [RFC2608]. However, SLPv2 authentication as defined in [RFC2608] does not provide a way to authenticate "zero result responses", leaving SLPv2 vulnerable to a denial of service attack. Such an attack can be carried out on a UA by sending it a "zero results" SrvRply or AttrRply, sent from a source address corresponding to a DA issuing a legitimate DAAdvert.

In addition, SLPv2 security as defined in [RFC2608] does not support confidentiality. When IPsec with ESP and a non-null transform is used to protect SLPv2, not only can unicast requests and replies be authenticated, but confidentiality can also be provided. This includes unicast requests to DAs and SAs as well as replies. It is also possible to actively discover SAs using multicast SA discovery, and then to send unicast requests to the discovered SAs.

As a result, for use with IP block storage protocols, it is believed that use of IPsec for security is more appropriate than the SLPv2 security model defined in [RFC2608].

Using IPsec to secure SLPv2 has performance implications. Security associations established between:

- UAs and SAs may be reused (the client on the UA host will use the service on the SA host).

- SAs and DAs may be reused (the SAs will reregister services)

- UAs and DAs will probably not be reused (many idle security associations are likely to result, and build up on the DA).

When IPsec is used to protect SLPv2, it is not necessarily appropriate for all hosts with whom an IPsec security association can be established to be trusted to originate SLPv2 service advertisements. This is particularly the case in environments where it is easy to obtain certificates valid for use with IPsec (for example, where anyone with access to the network can obtain a machine certificate valid for use with IPsec). If not all hosts are authorized to originate service advertisements, then it is necessary to distinguish between authorized and unauthorized hosts.

This can be accomplished by the following mechanisms:

[1] Configuring SAs with the identities or certificate characteristics of valid DAs, and configuring DAs with the identities of SAs allowed to advertise IP block storage services. The DAs are then trusted to enforce policies on service registration. This approach involves manual configuration, but avoids certificate customization for SLPv2.

[2] Restricting the issuance of certificates valid for use in SLPv2 service advertisement. While all certificates allowed for use with IPsec will chain to a trusted root, certificates for hosts authorized to originate service advertisements could be signed by an SLPv2-authorized CA, or could contain explicit SLPv2 authorizations within the certificate. After the IPsec security association is set up between the SLPv2 entities, the SLPv2 implementations can then retrieve the certificates used in the negotiation in order to determine whether the entities are authorized for the operations that are being performed. This approach requires less configuration, but requires some certificate customization for use with SLPv2.

2.6. iSNS security

The iSCSI protocol may use iSNS for discovery and management services, while the iFCP protocol is required to use iSNS for such services. In addition, iSNS can be used to store and distribute security policy and authorization information to iSCSI and iFCP devices. When the iSNS protocol is deployed, the interaction between iSNS server and iSNS clients are subject to the following additional security threats:

[1] An attacker can alter iSNS protocol messages, directing iSCSI and iFCP devices to establish connections with rogue devices, or weakening IPsec protection for iSCSI or iFCP traffic.

[2] An attacker can masquerade as the real iSNS server by sending false iSNS heartbeat messages. This could deceive iSCSI and iFCP devices into using rogue iSNS servers.

[3] An attacker can gain knowledge about iSCSI and iFCP devices by snooping iSNS protocol messages. Such information could aid an attacker in mounting a direct attack on iSCSI and iFCP devices, such as a denial-of-service attack or outright physical theft.

To address these threats, the following capabilities are needed:

[a] Unicast iSNS protocol messages may need to be authenticated. In addition, to protect against threat [3] above, confidentiality support is desirable, and REQUIRED when certain functions of iSNS are used.

[b] Multicast iSNS protocol messages such as the iSNS heartbeat message need to be authenticated. These messages need not be confidential since they do not leak critical information.

There is no requirement that the identities of iSNS entities be kept confidential. Specifically, the identity and location of the iSNS server need not be kept confidential.

In order to protect against an attacker masquerading as an iSNS server, client devices MUST support authentication of broadcast or multicast messages such as the iSNS heartbeat. The iSNS authentication block (which is identical in format to the SLP authentication block) MAY be used for this purpose. Note that the authentication block is used only for iSNS broadcast or multicast messages, and SHOULD NOT be used in unicast iSNS messages.

Since iSNS is used to distribute authorizations determining which client devices can communicate, IPsec authentication and data integrity MUST be supported. In addition, if iSNS is used to distribute security policy

for iFCP and iSCSI devices, then authentication, data integrity, and confidentiality MUST be supported and used.

Where iSNS is used without security, IP block storage protocol implementations MUST support a negative cache for authentication failures. This allows implementations to avoid continually contacting discovered endpoints which fail authentication within IPsec or at the application layer (in the case of iSCSI Login). The negative cache need not be maintained within the IPsec implementation, but rather within the IP block storage protocol implementation.

2.6.1. Use of iSNS to Discover Security Configuration of Peer Devices

In practice, within a single installation, iSCSI and/or iFCP devices may have different security settings. For example, some devices may be configured to initiate secure communication, while other devices may be configured to respond to a request for secure communication, but not to require security. Still other devices, while security capable, may neither initiate nor respond securely.

In practice, these variations in configuration can result in devices being unable to communicate with each other. For example, a device that is configured to always initiate secure communication will experience difficulties in communicating with a device that neither initiates nor responds securely.

The iSNS protocol is used to transfer naming, discovery, and management information between iSCSI devices, iFCP gateways, management stations, and the iSNS server. This includes the ability to enable discovery of security settings used for communication via the iSCSI and/or iFCP protocols.

The iSNS server stores security settings for each iSCSI and iFCP device interface. These security settings, which can be retrieved by authorized hosts, include use or non-use of IPsec, IKE, Main Mode, Aggressive Mode, PFS, Pre-shared Key, and certificates.

For example, IKE may not be enabled for a particular device interface. If a peer device can learn of this in advance by consulting the iSNS server, it will not need to waste time and resources attempting to initiate an IKE Phase 1 SA with that device interface.

If iSNS is used to distribute security policy, then the minimum information that should be learned from the iSNS server is the use or non-use of IKE and IPsec by each iFCP or iSCSI peer device interface. This information is encoded in the Security Bitmap field of each Portal of the peer device, and is applicable on a per-interface basis for the peer device. iSNS queries to acquire security configuration data about

peer devices MUST be protected by IPsec/ESP authentication.

2.6.2. Use of iSNS to Distribute iSCSI and iFCP Security Policies

Once communication between iSNS clients and the iSNS server are secured through use of IPsec, iSNS clients have the capability to discover the security settings required for communication via the iSCSI and/or iFCP protocols. Use of iSNS for distribution of security policies offers the potential to reduce the burden of manual device configuration, and decrease the probability of communications failures due to incompatible security policies. If iSNS is used to distribute security policies, then IPsec authentication, data integrity, and confidentiality MUST be used to protect all iSNS protocol messages.

The complete IKE/IPsec configuration of each iFCP and/or iSCSI device can be stored in the iSNS server, including policies that are used for IKE Phase 1 and Phase 2 negotiations between client devices. The IKE payload format includes a series of one or more proposals that the iSCSI or iFCP device will use when negotiating the appropriate IPsec policy to use to protect iSCSI or iFCP traffic.

Note that iSNS distribution of security policy is not necessary if the security settings can be determined by other means, such as manual configuration or IPsec security policy distribution. If an entity has already obtained its security configuration via other mechanisms, then it MUST NOT request security policy via iSNS.

For further details on how to store and retrieve IKE policy proposals in the iSNS server, see [iSNS].

2.6.3. iSNS Interaction with IKE and IPsec

When IPsec security is enabled, each iSNS client that is registered in the iSNS database maintains at least one Phase 1 and one Phase 2 security association with the iSNS server. All iSNS protocol messages between iSNS clients and the iSNS server are to be protected by a phase-2 security association.

2.6.4. iSNS Server Implementation Requirements

All iSNS implementations MUST support the replay protection mechanisms of IPsec. ESP in tunnel mode MUST be implemented, and IPsec with ESP in transport mode MAY be implemented.

To provide data origin authentication and integrity with ESP, HMAC-SHA1 MUST be supported, and AES in CBC MAC mode with XCBC extensions [AESXCBC] SHOULD be supported. When confidentiality is implemented, 3DES in CBC mode MUST be supported, and AES in Counter mode, as

described in [AESCTR], SHOULD be supported. DES in CBC mode SHOULD NOT be used due to its inherent weakness. If confidentiality is not required but data origin authentication and integrity is enabled, ESP with NULL Encryption MUST be used.

Conformant iSNS implementations MUST support IKE for authentication, negotiation of security associations, and key management, using the IPsec DOI, described in [RFC2407]. IP block storage protocols can be expected to send data in high volumes, thereby requiring rekey. Since manual keying does not provide rekeying support, its use is prohibited with IP block storage protocols. Although iSNS does not send a high volume of data, and therefore rekey is not a major concern, manual keying SHOULD NOT be used. This is for consistency, since dynamic keying support is already required in IP storage security implementations.

Conformant iSNS security implementations MUST support authentication using a pre- shared key, and MAY support certificate-based peer authentication using digital signatures. Peer authentication using the public key encryption methods outlined in [RFC2409] sections 5.2 and 5.3 SHOULD NOT be used.

Conformant iSNS implementations MUST support IKE Main Mode and SHOULD support Aggressive Mode. IKE Main Mode with pre-shared key authentication SHOULD NOT be used when either of the peers use dynamically assigned IP addresses. While Main Mode with pre-shared key authentication offers good security in many cases, situations where dynamically assigned addresses are used force use of a group pre-shared key, which is vulnerable to man-in-the-middle attack.

When digital signatures are used for authentication, either IKE Main Mode or IKE Aggressive Mode MAY be used. In all cases, access to locally stored secret information (pre-shared key or private key for digital signing) MUST be suitably restricted, since compromise of the secret information nullifies the security properties of the IKE/IPsec protocols.

When digital signatures are used to achieve authentication, an IKE negotiator SHOULD use IKE Certificate Request Payload(s) to specify the certificate authority (or authorities) that are trusted in accordance with its local policy. IKE negotiators SHOULD check the pertinent Certificate Revocation List (CRL) before accepting a PKI certificate for use in IKE's authentication procedures.

3. iSCSI security interoperability guidelines

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

3.1. iSCSI security issues

iSCSI provides for iSCSI Login, which includes support for application-layer authentication. This authentication is logically between the iSCSI Initiator and the iSCSI Target (as opposed to between the TCP/IP communication endpoints). The intent of the iSCSI design is that the Initiator and Target represent the systems (e.g., host and disk array or tape system) participating in the communication, as opposed to network communication interfaces or endpoints.

The iSCSI protocol, and iSCSI Login authentication do not meet the security requirements for iSCSI. iSCSI Login 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 Login authentication can mutually authenticates the Initiator to the Target, but does not by itself provide per-packet authentication, integrity, confidentiality or replay protection. In addition, iSCSI Login authentication, outlined in [iSCSI], does not provide for a protected ciphersuite negotiation. Therefore, iSCSI Login provides a weak security solution.

3.2. iSCSI and IPsec interaction

An iSCSI session [iSCSI], comprised of one or more TCP connections, is identified by the 2-tuple of the Initiator-defined identifier and the Target-defined identifier, <ISID, TSIH>. 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, Protocol (TCP), Source Port, Destination Port>. An IPsec Phase 2 SA is identified by the 3-tuple <Protocol (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.

When multiple iSCSI sessions are active between a given <Initiator, Target> pair, the set of TCP connections used by a given iSCSI session must be disjoint from those used by all other iSCSI sessions between the same <Initiator, Target> pair. Therefore a TCP connection can be associated with one and only one iSCSI session.

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 an IKE Phase 2 SA. The selectors may be specific to that TCP connection or may cover multiple connections. While each IKE Phase 2 SA may protect multiple TCP connections, 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 a TCP connection, an IKE Phase 2 SA and the iSCSI connection ID.

3.3. Initiating a New iSCSI Session

In order to create a new iSCSI Session, if an IKE Phase 1 SA does not already exist, then it is established by an Initiator implementing iSCSI security. Subsequent iSCSI connections established within the iSCSI session will typically be protected by IKE Phase 2 SAs derived from that IKE Phase 1 SA, although additional IKE Phase 1 SAs can also be brought up.

The Initiator and Target implementations successfully complete the IKE Phase 1 and Phase 2 negotiations before the iSCSI Initiator contacts the Target on well-known TCP port 3260, and sends the iSCSI Login command over the TCP connection. IPsec implementations configured with the correct policies (e.g. use ESP with non-null transform for all traffic destined for the iSCSI well-known TCP port 3260) will handle this automatically.

The Initiator fills in the ISID field, and leaves the TSIH 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, that identifies the TCP connection over which the Login command is being exchanged. When the iSCSI Target replies with its Login Command, both iSCSI devices will know the TSIH, and therefore the iSCSI session identifier <ISID, TSIH>.

A single iSCSI session identifier may have multiple associated IKE Phase 1 SAs, and each IKE Phase 1 SA may correspond to multiple iSCSI session identifiers. Each iSCSI connection (identified by the connection identifier) corresponds to a single TCP connection (identified by the 5-tuple). Each IKE Phase 2 SA is identified by the <Protocol (ESP), destination address, SPI> combination. A Phase 2 SA may protect multiple TCP connections, and corresponds to a single 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.

3.4. 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

When the iSCSI implementation wishes to close a session, it uses the appropriate iSCSI commands to accomplish this. After exchanging the appropriate iSCSI control messages for session closure, the iSCSI security implementation will typically initiate a half-close of each TCP connection within the iSCSI session.

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 [RFC793]. After the other side responds, the TIME WAIT state is entered. After the expiration of the TIME WAIT timeout, the TCP connection is closed.

3.5. Non-graceful iSCSI Teardown

If a given TCP connection unexpectedly fails, the associated iSCSI connection is torn down. There is no requirement that an IKE Phase 2 delete immediately follow iSCSI connection tear down or Phase 1 deletion. Since an IKE Phase 2 SA may correspond to multiple TCP connections, such a deletion might be inappropriate. Similarly, if the

IKE implementation receives a Phase 2 Delete message for a security association corresponding to a TCP connection, this does not necessarily imply that the TCP or iSCSI connection is to be torn down.

If a Logout Command/Logout Response sequence marks a connection for removal from the iSCSI session, then after the iSCSI peer has executed an iSCSI teardown process for the connection, the TCP connection will be closed. The iSCSI connection state can then be safely removed.

Since an IKE Phase 2 SA may be used by multiple TCP connections, an iSCSI implementation should not depend on receiving the IPsec Phase 2 delete as confirmation that the iSCSI peer has executed an iSCSI teardown process for the connection.

Since IPsec acceleration hardware may only be able to handle a limited number of active IKE Phase 2 SAs, Phase 2 delete messages may be sent for idle SAs, as a means of keeping the number of active Phase 2 SAs to a minimum. The receipt of an IKE Phase 2 delete message SHOULD NOT be interpreted as a reason for tearing down the corresponding iSCSI connection if no Logout Command/Logout Receive has been executed on the connection. Rather, it is preferable to leave the iSCSI connection up, and if additional traffic is sent on it, to bring up another IKE Phase 2 SA to protect it. This avoids the potential for continually bringing iSCSI connections up and down.

3.6. Application-layer CRC

iSCSI's error detection and recovery assumes that the TCP and IP checksums provide inadequate integrity protection for high speed communications. As described in [CRCTCP], when operating at high speeds, the 16-bit TCP checksum [RFC793] will not necessarily detect all errors, resulting in possible data corruption. iSCSI [iSCSI] therefore incorporates a 32-bit CRC to protect its headers and data.

When a CRC check fails (i.e. CRC computed at receiver does not match the received CRC), the iSCSI PDU covered by that CRC is 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 offers protection against an attacker attempting to modify packets in transit, as well as unintentional packet modifications caused by network malfunctions or other errors. In general, IPsec authentication transforms afford stronger integrity protection than both the 16-bit TCP checksum and the 32-bit application-layer CRC that is specified for use with iSCSI. 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.

3.6.1. Simplification of recovery logic

Where IPsec integrity protection is known to be in place, and covers the entire connection between iSCSI endpoints (or the portion that requires additional integrity connection), portions of iSCSI can be simplified. For example, mechanisms to recover from CRC check failures are not necessary.

If the iSCSI CRC is negotiated, the recovery logic can 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 very rare for errors undetected by IPsec integrity protection to be detected by the iSCSI CRC.

3.6.2. Omission of iSCSI CRC

In some situations where IPsec is employed, the iSCSI CRC will not provide additional protection, and can be omitted.

For example, where IPsec processing as well as TCP checksum and iSCSI CRC verification are offloaded within the NIC, each of these checks will be verified prior to transferring data across the bus, so that subsequent errors will not be detected. As a result, where IPsec processing is offloaded to the NIC, the iSCSI CRC is not necessary and the implementations may wish not to negotiate it.

However, in other circumstances, the TCP checksum and iSCSI CRC will provide additional protection, and it is desirable to negotiate use of the iSCSI CRC even though IPsec is available. These situations occur where:

[1] IPsec, TCP and iSCSI are implemented purely in software. Here, additional failure modes may be detected by the TCP checksum and/or iSCSI CRC. For example, after the IPsec message integrity check is successfully verified, the segment is copied as part of TCP processing, and a memory error during this process might cause the TCP checksum or iSCSI CRC verification to fail.

[2] The implementation is an iSCSI-iSCSI proxy or gateway. Here the iSCSI CRC can be propagated 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.

[3] IPsec is provided by a device external to the actual iSCSI device. Here 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 to protect against NIC and network errors, although it may be desirable for other reasons (e.g., [1] and [2] above).

4. iFCP and FCIP security issues

4.1. iFCP and FCIP Authentication Requirements

iFCP and FCIP are peer-to-peer protocols. iFCP and FCIP sessions may be initiated by either or both peer gateways. Consequently, bi-directional authentication of peer gateways MUST be provided.

iFCP and FCIP are transport protocols that encapsulate SCSI and Fibre Channel frames over IP. Therefore, Fibre Channel, operating system, and user identities are transparent to the iFCP and FCIP protocols.

iFCP gateways use Discovery Domain information obtained from the iSNS server to determine whether the initiating Fibre Channel N_PORT should be allowed access to the Target N_PORT. N_PORT identities used in the Port Login (PLOGI) process will be considered authenticated provided that they are received over a connection whose security complies with the local security policy.

There is no requirement that the identities used in authentication be kept confidential.

4.2. iFCP Interaction with IPsec and IKE

A conformant iFCP Portal is capable of establishing one or more IKE Phase-1 Security Associations (SAs) to a peer iFCP Portal. A Phase-1 SA may be established when an iFCP Portal is initialized, or may be deferred until the first TCP connection with security requirements is established.

An IKE Phase-2 SA protects one or more TCP connections within the same iFCP Portal. More specifically, the successful establishment of an IKE Phase-2 SA results in the creation of two uni-directional IPsec SAs

fully qualified by the tuple <SPI, destination IP address, ESP>. These SAs protect the setup process of the underlying TCP connections and all their subsequent TCP traffic. Each of the TCP connections protected by an SA is either in the unbound state, or is bound to a specific iFCP session.

In summary, at any point in time:

[1] There exist 0..M IKE Phase-1 SAs between peer iFCP portals
[2] Each IKE Phase-1 SAs has 0..N IKE Phase-2 SAs
[3] Each IKE Phase-2 SA protects 0..Z TCP connections

The creation of an IKE Phase-2 SA may be triggered by security policy rules retrieved from an iSNS server. Alternately, the creation of an SA may be triggered by policy rules configured through a management interface, reflecting iSNS-resident policy rules. Likewise, the use of a Key Exchange payload in Quick Mode for perfect forward secrecy may be driven by security policy rules retrieved from the iSNS server, or set through a management interface.

If an iFCP implementation makes use of unbound TCP connections, and such connections belong to an iFCP Portal with security requirements, then the unbound connections MUST be protected by an SA at all times just like bounded connections.

Upon receiving an IKE Phase-2 delete message, there is no requirement to terminate the protected TCP connections or delete the associated IKE Phase-1 SA. Since an IKE Phase-2 SA may be associated with multiple TCP connections, terminating such connections might in fact be inappropriate and untimely.

To minimize the number of active Phase-2 SAs, IKE Phase-2 delete messages may be sent for Phase-2 SAs whose TCP connections have not handled data traffic for a while. To minimize the use of SA resources while the associated TCP connections are idle, creation of a new SA should be deferred until new data are to be sent over the connections.

4.3. FCIP Interaction with IPsec and IKE

FCIP Entities establish tunnels with other FCIP Entities in order to transfer IP encapsulated FC frames. Each tunnel is a separate FCIP Link, and can encapsulate multiple TCP connections. The binding of TCP connections to an FCIP Link is performed using the Fibre Channel World Wide Names (WWNs) of the two FCIP Entities.

FCIP Entities may have more than one interface and IP address, and it is possible for an FCIP Link to contain multiple TCP connections whose FCIP endpoint IP Addresses are different. In this case, an IKE Phase 1 SA is

typically established for each FCIP endpoint IP Address pair. For the purposes of establishing an IKE Phase 1 SA, static IP addresses are typically used for identification.

Each TCP connection within an FCIP Link corresponds to an IKE Phase 2 (Quick Mode) SA. This is established prior to sending the initial TCP SYN packet and applies to the life of the connection. Phase 2 negotiation is also required for rekeying, in order to protect against replay attacks.

FCIP implementations MAY provide administrative management of Confidentiality usage. These management interfaces SHOULD be provided in a secure manner, so as to prevent an attacker from subverting the security process by attacking the management interface.

FCIP Entities do not require any user-level authentication, since all FCIP Entities participate in a machine-level tunnel function. FCIP uses SLP for discovery, but not to distribute security policies.

5. Security considerations

5.1. Transport mode versus tunnel mode

With respect to block storage protocols, the major differences between the IPsec tunnel mode and transport mode are as follows:

[1] MTU considerations. Tunnel mode introduces an additional IP header into the datagram that reflects itself in a corresponding decrease in the path MTU seen by packets traversing the tunnel. This may result in a decrease in the maximum segment size of TCP connections running over the tunnel.

[2] Address assignment and configuration. Within IPsec tunnel mode, it is necessary for inner and outer source addresses to be configured, and for inner and outer destination addresses to be discovered. Within transport mode it is only necessary to discover a single destination address and configure a single source address. IPsec tunnel mode address usage considerations are discussed in more detail below.

[3] NAT traversal. As noted in [IPsecNATReq], IPsec tunnel mode ESP can traverse NAT in limited circumstances, whereas transport mode ESP cannot traverse NAT. To enable NAT traversal in the general case, the IPsec NAT traversal functionality described in [UDPIPsec], [IPsecNATJust], [NATIKE] can be implemented. More details are provided in the next section.

[4] Firewall traversal. Where a block storage protocol is to traverse administrative domains, the firewall administrator may desire to verify the integrity and authenticity of each transiting packet, rather than opening a hole in the firewall for the block storage protocol. IPsec tunnel mode lends itself to such verification, since the firewall can terminate the tunnel mode connection while still allowing the endpoints to communicate end-to-end. If desired, the endpoints can in addition utilize IPsec transport mode for end-to-end security, so that they can also verify authenticity and integrity of each packet for themselves.

In contrast, carrying out this verification with IPsec transport mode is more complex, since the firewall will need to terminate the IPsec transport mode connection and will need to act as an iSCSI, iFCP or FCIP gateway or TCP proxy, originating a new IPsec transport mode connection from the firewall to the internal endpoint. Such an implementation cannot provide end-to-end authenticity or integrity protection, and an application-layer CRC (iSCSI) or forwarding of the Fibre Channel frame CRC (iFCP and FCIP) is necessary to protect against errors introduced by the firewall.

[5] IPsec-application integration. Where IPsec and the application layer protocol are implemented on the same device or host, it is possible to enable tight integration between them. For example, the application layer can request and verify that connections are protected by IPsec, and can obtain attributes of the IPsec security association. While in transport mode implementations the IPsec and application protocol implementations typically reside on the same host, with IPsec tunnel mode, they may reside on different hosts. Where IPsec is implemented on an external gateway, integration between the application and IPsec is typically not possible. This limits the ability of the application to control and verify IPsec behavior.

5.1.1. IPsec tunnel mode addressing considerations

IPsec tunnels include both inner and outer source as well as destination addresses.

When used with IP block storage protocols, the inner destination address represents the address of the IP block storage protocol peer (e.g. the ultimate destination for the packet). The inner destination address can be discovered using SLPv2 or iSNS, or can be resolved from an FQDN via DNS, so that obtaining this address is typically not an issue.

The outer destination address represents the address of the IPsec security gateway used to reach the peer. Several mechanisms have been

proposed for discovering the IPsec security gateway used to reach a particular peer. [RFC2320] discusses the use of KX Resource Records (RRs) for IPsec gateway discovery. However, while KX RRs are supported by many DNS server implementations, they have not yet been widely deployed. Alternatively, DNS SRV [RFC2782] can also be used for this purpose, as can protocols such as Tunnel Endpoint Discovery [TED].

When used with IP block storage protocols, the outer source address is configured either statically or dynamically, using mechanisms such as DHCPv4 [RFC2131], DHCPV6 [DHCPv6], or stateless address autoconfiguration [RFC2373].

The inner source address SHOULD be included in the Quick Mode ID payload when the peer establishes a tunnel mode SA with the IPsec security gateway. This enables the IPsec security gateway to properly route packets back to the remote peer. The inner source address can be configured via a variety of mechanisms, depending on the scenario. Where the IP block storage peers are located within the same administrative domain, it is typically possible for the inner and outer source addresses to be the same. This will work because the outer source address is presumably assigned from within a prefix assigned to the administrative domain, and which is therefore routable within the domain. Assuming that the IPsec security gateway is aware of the inner source address used by the connecting peer and plumbs a host route for it, then packets arriving at the IPsec security gateway destined for the address can be correctly encapsulated and sent down the correct tunnel.

Where IP block storage peers are located within different administrative domains, it may be necessary for the inner source address to be assigned by the IPsec security gateway, effectively "joining" the remote host to the LAN attached to the IPsec security gateway. For example, if the remote peer were to use its assigned (outer) source address as the inner source address, then a number of problems might result:

[1] Intrusion detection systems sniffing the LAN behind the IPsec security gateway would notice source addresses originating outside the administrative domain.

[2] Reply packets might not reach their destination, since the IPsec security gateway typically does not advertise default, but rather only the prefix that it allocates addresses from. Since the remote peer's address does not originate from with a prefix native to the administrative domain, it is likely that routers within the domain would not have a route for it, and would send the packet off to the router advertising default (perhaps a border router), instead of to the IPsec security gateway.

For these reasons, for inter-domain use, assignment of inner source addresses may be needed. At present this is not a very common scenario; however, where this is necessary, DHCP-based address assignment within IPsec tunnel mode [DHCPIPsec] MUST be supported. Note that this mechanism is not yet widely deployed within IPsec security gateways; existing IPsec tunnel mode servers typically implement this functionality via proprietary extensions to IKE.

5.2. NAT traversal

As noted in [IPsecNATJust], tunnel mode ESP can traverse NAT in a limited set of circumstances. For example, if there is only one protocol endpoint behind a NAT, "ANY to ANY" selectors are negotiated, and the receiver does not perform source address validation, then tunnel mode ESP may successfully traverse NATs. Since ignoring source address validation introduces new security vulnerabilities, and negotiation of specific selectors is desirable so as to limit the traffic that can be sent down the tunnel, these conditions may not necessarily apply, and tunnel mode NAT traversal will not always be possible.

TCP carried within 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 ESP message integrity check. As a result, the TCP checksum will be invalidated by a NAT located between the two endpoints.

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 while remaining in compliance, iSCSI, iFCP or FCIP security implementations can implement IPsec/IKE NAT traversal, as described in [IPsecNATReq], [UDPIPsec], [IPsecNATJust], [NATIKE].

The IKE [NATIKE] and IPsec [UDPIPsec] NAT traversal specifications enable 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.

5.3. IKE issues

There are situations where it is necessary for IKE to be implemented in firmware. In such situations, it is important to keep the size of the IKE implementation within strict limits. 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 5.3-1 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, implementations typically will not require this in order to fit within the code size budget.

5.4. Rekeying issues

It is expected that IP block storage 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, a single IPsec SA could rapidly cycle through the 32-bit IPsec sequence number space.

For example, a single SA operating at 1 Gbps line rate and sending 64 octet packets 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. At 10 Gbps, exhaustion would occur in 220 seconds or **3.67 minutes. With 1518 octet packets, this would occur within 1.45** hours.

In the future, it may be desirable for implementations operating at speeds of 1 Gbps or greater to implement IPsec sequence number extension, described in [Seq]. Note that depending on the transform in use, it is possible that rekeying will be required prior to exhaustion of the sequence number space.

In CBC-mode ciphers the ciphertext of one block depends on the plaintext of that block as well as the ciphertext of the previous block. This implies that if the ciphertext of two blocks are identical, and the plaintext of the next block is also identical, then the ciphertext of the next block will be identical. Thus, if identical ciphertext blocks can be found with matching subsequent blocks, an attacker can determine the existence of matching plaintext.

Such "Birthday attacks" were examined by Bellare et. al. in [DESANALY]. On average, a ciphertext block of size n bits will be expected to repeat every 2^[n/2] blocks. Although a single "birthday attack" does not provide much information to an attacker, multiple such attacks might provide useful information. To make this unlikely, it is recommended that a rekey occur before 2^[n/2] blocks have been sent on a given SA. These conclusions do not apply to counter mode. Bellare et. al. have formalized this in [DESANALY], 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. These conclusions do not apply to counter mode.

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 5.3-1 - Code Size for IKE implementations

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

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

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 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.

5.5. Transform issues

Since IP block storage 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 or slower.

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.

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
>
> "

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

The MD5 algorithm is specified in [MD5]; HMAC is defined in [RFC2104] and security issues with MD5 are discussed in [MD5Attack]. The HMAC-MD5 IPsec transform is specified in [HMACMD5IPsec]. The HMAC-SHA1 IPsec transform is specified in [RFC2404].

In addition to these existing algorithms, NIST is currently evaluating the following modes [NSPUE2] of AES, defined in [RIJNDAEL],[NISTAES]:

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 utilizing AES modes, it may be necessary to use larger public keys; the tradeoffs are described in [KeyLen]. Additional MODP Diffie-Hellman groups for use with IKE are described in [MODP].

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

PMAC

To provide authentication, integrity and replay protection, IP block storage security implementations MUST support HMAC-SHA1 [RFC2404] for authentication, and AES in CBC MAC mode with XCBC extensions SHOULD be supported [AESXCBC].

HMAC-SHA1 [RFC2404] is to be preferred to HMAC-MD5, due to concerns that have been raised about the security of MD5 [MD5Attack]. 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 [RFC2404] exists. As a result, it is most practical to utilize HMAC-SHA1 as the authentication algorithm for products shipping in the near future. The HMAC-SHA2 algorithm [NISTSHA] is also under development, including an IPsec transform [SHAEXT], so that this may merit consideration in the future. AES in CBC-MAC authentication mode with XCBC extensions was selected since it avoids adding substantial additional code if AES is already being implemented for encryption; an IPsec transform document is available [AESCBC].

Authentication transforms also exist that are considerably more efficient to implement than HMAC-SHA1, HMAC-SHA2 or AES in CBC-MAC authentication mode. UMAC [UMAC],[UMACKR] is more efficient to implement in software and PMAC [PMAC] is more efficient to implement in hardware. PMAC is currently being evaluated as part of the NIST modes effort [MODES] but an IPsec transform does not yet exist; neither does a

UMAC transform.

For confidentiality, the ESP mandatory-to-implement algorithm (DES) is unacceptable. As noted in [DESCRACK], DES is crackable with modest computation resources, and so is inappropriate for use in situations requiring high levels of security.

To provide confidentiality for iSCSI, iFCP, and FCIP, 3DES in CBC mode [RFC2451] MUST be supported and AES in Counter Mode [AESCTR] SHOULD be supported. For use in high speed implementations, 3DES has significant disadvantages. However, a 3DES IPsec transform has been specified and parts are available that can run at 1 Gbps, implementing 3DES in products is practical for the short term.

As described in Appendix B, 3DES software implementations make excessive computational demands at speeds of 100 Mbps or greater, effectively ruling out software-only implementations. 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 faster. Thus, 3DES is inconvenient for use at very high speeds, as well as being impractical for implementation in software at slower speeds (100+ Mbps).

5.6. Fragmentation Issues

When certificate authentication is used, IKE fragmentation can be encountered. This can occur when certificate chains are used, or even when exchanging a single certificate if the key size, or size of other certificate fields (such as the distinguished name and other OIDs) is large enough. Many Network Address Translators (NATs), and firewalls do not handle fragments properly, dropping them on inbound and/or outbound.

Routers in the path will also frequently discard fragments after the initial one, since they typically will not contain full IP headers that can be compared against an Access List.

As a result, where IKE fragmentation occurs, the endpoints will often be unable to establish an IPsec security association. The solution to this problem is to install NAT, firewall or router code load that can properly support fragments. If this cannot be done, then the following alternatives can be considered:

[1] Obtaining certificates by other means.

[2] Reducing the size of the certificate chain.

[3] Reducing the size of fields within the certificates. This includes reduction in the key size, the distinguished name or other fields. This should be considered only as a last resort.

Fragmentation can become a concern when prepending IPsec headers to a frame. One mechanism which can be used to reduce this problem is to utilize path MTU discovery. For example, when TCP is used as the transport for iSCSI, iFCP or FCIP then path MTU discovery, described in [RFC1191], [RFC1435], [RFC1981], 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 [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 ICMP PMTU is received by an IPsec implementation that processes unauthenticated ICMP packets, this value should be stored in the SA as proposed in [RFC2401], and IPsec should also provide notification of this event to TCP so that the new MTU value can be correctly reflected.

5.7. Security Checks

When a connection is opened which requires security, IP block storage security implementations may wish to check that the connection is protected by IPsec. If security is desired and IPsec protection is removed on a connection, it is reinstated before non-protected IP block storage packets are sent. Since IPsec verifies that each packet arrives on the correct SA, as long as it can be ensured that IPsec protection is in place, then IP block storage implementations can be assured that each received packet was sent by a trusted peer.

When used with IP block storage protocols, each TCP connection MUST be protected by an IKE Phase 2 SA. Traffic from one or more than one TCP connection may flow within each IPsec Phase 2 SA. IP block storage 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 connection. This check will be performed by IPsec, preventing malicious peers from sending commands on inappropriate Quick Mode SAs.

5.8. Authentication issues

5.8.1. 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 entity. 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. For iFCP and FCIP, the certificate credentials provided will almost always be those of the device and will be stored on the device.

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 login (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. Since iFCP and FCIP have no equivalent of iSCSI Login, for these protocols only the machine is authenticated.

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.

5.8.2. 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 hosts (such as with iSCSI 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 pre-shared key authentication is used in Main Mode along with entities whose address is dynamically assigned, the same pre-shared key is shared by a group and is no longer able to function as an effective shared secret. In this situation, neither the Initiator nor Responder 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 is typically not of concern where IP addresses are typically statically assigned (such as with iFCP and FCIP), since in this situation individual pre-shared keys are possible within IKE Main Mode.

However, where IP addresses are dynamically assigned and Main Mode is used along with pre-shared keys, the Responder is not authenticated unless application-layer mutual authentication is performed (e.g. iSCSI login authentication). This enables an attacker in possession of the group pre-shared key to masquerade as the Responder. In addition to enabling the attacker to present false data, the attacker would also be able to mount a dictionary attack on legacy authentication methods such as CHAP [RFC1994], potentially compromising many passwords at a time. This vulnerability is widely present in existing IPsec implementations.

Group pre-shared keys are not required 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.

Note that care needs to be taken with IKE Phase 1 Identity Payload selection in order to enable mapping of identities to pre-shared keys even with Aggressive Mode. Where the ID_IPV4_ADDR or ID_IPV6_ADDR Identity Payloads are used and addresses are dynamically assigned, mapping of identities to keys is not possible, so that group pre-shared keys are still a practical necessity. As a result, identities other than ID_IPV4_ADDR and ID_IPV6_ADDR (such as ID_FQDN or ID_USER_FQDN) SHOULD be employed in situations where Aggressive mode is utilized along with pre-shared keys and IP addresses are dynamically assigned.

5.8.3. IKE and application-layer authentication

In addition to IKE authentication, iSCSI implementations utilize their own authentication methods. Currently, work is underway on Fibre Channel security, so that a similar authentication process may eventually also apply to iFCP and FCIP as well.

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 Login 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 Login 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 Login 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 IPsec SA is opened, another user on a multi-user machine may be able to send traffic down the IPsec SA. This is true for both transport mode and tunnel mode SAs.

To limit the potential vulnerability, IP block storage implementations MUST do the following:

[1] Ensure that socket access is appropriately controlled. On a multi-user operating system, this implies that sockets opened for use by IP block storage protocols MUST be exclusive.

[2] In the case of iSCSI, implementations MUST also ensure that application layer login credentials (such as iSCSI login credentials) are protected from unauthorized access.

If these directives are followed, then a rogue process will not be able to access an IP block storage volume.

While the identity asserted within IKE is verified on a per-packet basis, to avoid interference between users on a given machine, operating system support is required. In order to segregate traffic between users

when user authentication is supported, the IPsec endpoints must ensure that only traffic from that particular user is sent or received within the IPsec SA. Enforcement of this restriction is the responsibility of the operating system.

In kernel mode iSCSI drivers there typically is no user context to perform user authentication. In this case the authentication is closer to machine authentication. In most operating systems device permissions would control the ability to read/write to the device prior to mounting. Once the device is mounted, ACLs set by the filesystem control access to the device. An administrator can access the blocks on the device directly (for instance, by sending pass through requests to the port driver directly such as in Windows NT). In the same way, an administrator can open a raw socket and send IPsec protected packets to an iSCSI Target. The situation is analogous, and in this respect no new vulnerability is created that didn't previously exist. The Operating system's ACLs need to be effective to protect a device (whether it is a SCSI device or an iSCSI device).

5.8.4. IP block storage authorization

IP block storage protocols can use a variety of mechanisms for authorization. Where ID_FQDN is used as the Identity Payload, IP block storage endpoints can be configured with a list of authorized FQDNs. The configuration can occur manually, or automatically via iSNS or the iSCSI MIB, defined in [AuthMIB].

Assuming that IPsec authentication is successful, this list of FQDNs can be examined to determine authorization levels. Where certificate authentication is used, it is possible to configure IP block storage protocol endpoints with trusted roots. The trusted roots used with IP block storage protocols might be different from the trusted roots used for other purposes. If this is the case, then the burden of negotiating use of the proper certificates lies with the IPsec initiator.

Note that because IKE does not deal well with certificate chains, and is typically configured with a limited set of trusted roots, it is most appropriate for intra-domain usage.

Since iSCSI supports Login authentication, it is also possible to use the identities presented within the iSCSI Login for authorization purposes.

In iFCP, basic access control properties stem from the requirement that two communicating iFCP gateways be known to one or more iSNS servers before they can engage in iFCP exchanges. The optional use of discovery domains in iSNS yields access control schemas of greater complexity.

5.9. Use of AES in counter mode

When utilizing AES modes, it may be necessary to use larger public keys; the tradeoffs are described in [KeyLen]. Additional MODP Diffie-Hellman groups for use with IKE are described in [MODP].

When AES in counter mode is used, it is important to avoid reuse of the counter with the same key, even across all time. Counter mode creates ciphertext by XORing generated key stream with plaintext. It is fairly easy to recover the plaintext from two messages counter mode encrypted under the same counter value, simply by XORing together the two packets. The implication of this is that it is an error to use IPsec manual keying with counter mode, except when the implementation takes heroic measures to maintain state across sessions. In any case, manual keying MUST NOT be used since it does not provide the necessary rekeying support.

Another counter mode issue is it makes forgery of correct packets trivial. Counter mode should therefore never be used without also using data authentication.

6. Normative references

[RFC793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.

[RFC1191] Mogul, J., and S. Deering, "Path MTU Discovery", RFC 1191, November 1990

[RFC1435] Knowles, S., "IESG Advice from Experience with Path MTU Discovery", RFC 1435, March 1993

[RFC1981] McCann, J., Deering, S. and J. Mogul, "Path MTU Discovery for IP version 6", RFC 1981, August 1996

[RFC2104] Krawczyk, H., Bellare, M. and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997

[RFC2401] Atkinson, R. and Kent, S., "Security Architecture for the Internet Protocol", RFC 2401, November 1998

[RFC2404] Madson, C., Glenn, R., "The Use of HMAC-SHA-1-96 within ESP and AH", RFC 2404, November 1998

[RFC2406] Kent, S., Atkinson, R., "IP Encapsulating Security Payload (ESP)", RFC 2406, November 1998

[RFC2407] Piper, D., "The Internet IP Security Domain of Interpretation of ISAKMP", RFC 2407, November 1998

[RFC2408] Maughan, D., Schertler, M., Schneider, M., Turner, J., "Internet Security Association and Key Management Protocol (ISAKMP), RFC 2408, November 1998

[RFC2409] Harkins, D., Carrel, D., "The Internet Key Exchange (IKE)", RFC 2409, November 1998

[RFC2412] Orman, H., "The OAKLEY Key Determination Protocol", RFC 2412, November 1998

[RFC2451] Pereira, R., Adams, R., "The ESP CBC-Mode Cipher Algorithms", RFC 2451, November 1998

[RFC2608] Guttman, E., Perkins, C., Veizades, J., Day, M, "Service Location Protocol, Version 2", RFC 2608, June 1999

[RFC2923] Lahey, K., TCP Problems with Path MTU Discovery", RFC 2923, September 2000

[RFC2945] Wu, T., "The SRP Authentication and Key Exchange System," RFC 2945, September 2000

[NISTAES] Draft FIPS Publication ZZZZ, "Advanced Encryption Standard (AES)", U.S. DoC/NIST, summer 2001

[3DESANSI] 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

[AESXCBC] Frankel, S., Herbert, H., "The AES-XCBC-MAC-96 Algorithm and Its Use with IPsec", Internet draft (work in progress), draft-ietf-ipsec-ciph-aes-xcbc-mac-01.txt, May 2002.

[AESCTR] Walker, J., Moskowitz, R., "The AES128 CTR Mode of Operation and Its Use with IPsec", Internet draft (work in progress), draft-moskowitz-aes128-ctr-00.txt, September 2001

[DHCPIPsec] Patel, B., Aboba, B., Kelly, S., Gupta, V., "DHCPv4 Configuration of IPsec Tunnel Mode", Internet draft (work in progress), draft-ietf-ipsec-dhcp-13.txt, June 2001

[iSCSI] Satran, J., et al., "iSCSI", Internet draft (work in progress), draft-ietf-ips-iSCSI-14.txt, July 2002

[iFCP] Monia, C., et al., "iFCP - A Protocol for Internet Fibre Channel Storage Networking", Internet drafts (work in progress), draft-ietf-ips-ifcp-12.txt, June 2002

[FCIP] Rajagopal, M., et al., "Fibre Channel over TCP/IP (FCIP)", Internet draft (work in progress), draft-ietf-ips-fcovertcpip-11.txt, June 2002.

[iSCSIName] Bakke, M., et al., "iSCSI Naming and Discovery", draft-ietf-ips-iscsi-name-disc-06.txt, Work in Progress, June 2002

[FCIPSLP] Petersen, D., "Finding FCIP Entities Using SLP", Internet draft (work in progress), draft-ietf-ips-fcip-slp-01.txt, November 2001

[iSCSISLP] Bakke, M., "Finding iSCSI Targets and Name Servers Using SLP", Internet draft (work in progress), draft-ietf-ips-iscsi-slp-03.txt, March 2002

[iSNS] Gibbons, K., et al., "iSNS Internet Storage Name Service", Internet draft (work in progress), draft-ietf-ips-isns-11.txt, June 2002

[SRPNDSS] Wu, T., "The Secure Remote Password Protocol", in Proceedings of the 1998 Internet Society Symposium on Network and Distributed Systems Security, San Diego, CA, pp. 97-111

[KeyLen] Orman, H., Hoffman, P., "Determining Strengths For Public Keys Used For Exchanging Symmetric Keys", Internet draft (work in progress), draft-orman-public-key-lengths-05.txt, December 2001.

[MODP] Kivinen, T., Kojo, M., "More MODP Diffie-Hellman groups for IKE", Internet draft (work in progress), draft-ietf-ipsec-ike-modp-groups-04.txt, December 2001.

7. Informative references

[RFC2373] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 2373, July 1998.

[RFC2402] Kent, S., Atkinson, R., "IP Authentication Header", RFC 2402, November 1998

[RFC2782] Gulbrandsen, A., Vixie, P., Esibov, L. "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, February 2000.

[RFC2865] Rigney, C., Willens, S., Rubens, A. and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, June 2000.

[RFC2983] Black, D. "Differentiated Services and Tunnels", RFC 2983, October 2000.

[AuthMIB] Bakke, M., et al., "Definitions of Managed Objects for iSCSI", Internet draft (work in progress), draft-ietf-ips-iscsi-mib-03.txt, October 2001.

[MD5] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, April 1992

[MD5Attack] Dobbertin, H., "The Status of MD5 After a Recent Attack", CryptoBytes Vol.2 No.2, Summer 1996

[CRCTCP] Stone J., Partridge, C., "When the CRC and TCP checksum disagree", ACM Sigcomm, Sept. 2000

[DESCRACK] Cracking DES, O'Reilly & Associates, Sebastapol, CA 2000

[iSCSIREQ] Krueger, M., et al., "iSCSI Requirements and Design Considerations", draft-ietf-ips-iscsi-reqmts-06.txt, Work in Progress, March 2002

[IPsecNatReq] Aboba, B., "IPsec-NAT Compatibility Requirements", draft-ietf-ipsec-nat-reqts-01.txt, Work in Progress, March 2002.

[UDPIPsec] Huttunen, A. et. al., "UDP Encapsulation of IPsec Packets", draft-ietf-ipsec-udp-encaps-01.txt, October 2001

[Seq] Kent, S., "IP Encapsulating Security Payload (ESP)", Internet draft (work in progress), draft-ietf-ipsec-esp-v3-01.txt, November 2002

[IPsecNATJust] Dixon, W. et. al., "IPsec over NAT Justification for UDP Encapsulation", draft-ietf-ipsec-udp-encaps-justification-00.txt, June 2001

[NATIKE] Kivinen, T., et al., "Negotiation of NAT-Traversal in the IKE", Internet draft (work in progress), draft-ietf-ipsec-

nat-t-ike-01.txt, October 2001

[HMACMD5IPsec]
Madson, C., Glenn, R., "The Use of HMAC-MD5-96 within ESP and AH", RFC 2403, November 1998

[PMAC] 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

[TED] Fluhrer, S., "Tunnel Endpoint Discovery", Internet draft (work in progress), draft-fluhrer-ted-00.txt, November 2001.

[UMAC] 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

[NISTSHA] "Descriptions of SHA-256, SHA-384, and SHA-512," http://csrc.nist.gov/cryptval/shs/sha256-384-512.pdf

[FIPS46-3] U.S. DoC/NIST, "Data encryption standard (DES)", FIPS 46-3, October 25, 1999

[FIPS74] U.S. DoC/NIST, "Guidelines for implementing and using the nbs data encryption standard", FIPS 74, Apr 1981

[DESDIFF] Biham, E., Shamir, A., "Differential Cryptanalysis of DES-like cryptosystems", Journal of Cryptology Vol 4, Jan 1991

[RFC2405] Madson, C., Doraswamy, N., "The ESP DES-CBC Cipher Algorithm With Explicit IV", RFC 2405, November 1998

[RIJNDAEL] Daemen, J., Rijman, V., "AES Proposal: Rijndael," NIST AES Proposal, June 1998 http://csrc.nist.gov/encryption/aes/round2/AESAlgs/Rijndael/Rijndael.pdf

[MODES] "Symmetric Key Block Cipher Modes of Operation," http://www.nist.gov/modes

[NSPUE2] "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

[AESOCB] Moskowitz, R., Walker, J., "The AES128 OCB Mode of Operation and Its Use with IPsec", Internet draft (work in progress), draft-moskowitz-aes128-ocb-00.txt, September 2001

[CTR-MODE] Lipmaa, H., Rogaway, P., Wagner, D., "CTR-MODE encryption", Comment on mode of operations NIST, Jan 2001

[AESPERF] 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

[PENTPERF] A. Bosselaers, "Performance of Pentium implementations", http://www.esat.kuleuven.ac.be/~bosselae/

[UMACPERF] Rogaway, P., "UMAC Performance", http://www.cs.ucdavis.edu/~rogaway/umac/perf00.html

[DESINT] Bellovin, S., "An Issue With DES-CBC When Used Without Strong Integrity", Proceedings of the 32nd IETF, Danvers, MA, April 1995

[UMACKR] 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. Also available at: http://www.cs.ucdavis.edu/~rogaway/umac/draft-krovetz-umac-01.txt

[RFC1994] Simpson, W.,"PPP Challenge Handshake Authentication Protocol (CHAP)," RFC 1994, August 1996.

[DESANALY] Bellare, Desai, Jokippi, Rogaway, "A Concrete Treatment of Symmetric Encryption: Analysis of the DES Modes of Operation", 1997, http://www-cse.ucsd.edu/users/mihir/

[SRPDIST] Wu, T., "SRP Distribution", http://www-cs-students.stanford.edu/~tjw/srp/download.html

[SHAEXT] Frankel, S., Kelly, S., "The Use of SHA-256, SHA-384, and SHA-512 within ESP, AH and IKE," Work in progress

Appendix A - Well Known Groups for Use with SRP

Modulus (N) and generator (g) values for various modulus lengths are given below. The values below are taken from software developed by Tom Wu and Eugene Jhong for the Stanford SRP distribution [SRPDIST], and subsequently rigorously verified to be prime:

[768 bits]
Modulus (base 16) =
B344C7C4F8C495031BB4E04FF8F84EE95008163940B9558276744D91F7CC9F40
2653BE7147F00F576B93754BCDDF71B636F2099E6FFF90E79575F3D0DE694AFF
737D9BE9713CEF8D837ADA6380B1093E94B6A529A8C6C2BE33E0867C60C3262B
Generator = 2

[1024 bits]
Modulus (base 16) =
EEAF0AB9ADB38DD69C33F80AFA8FC5E86072618775FF3C0B9EA2314C9C256576
D674DF7496EA81D3383B4813D692C6E0E0D5D8E250B98BE48E495C1D6089DAD1
5DC7D7B46154D6B6CE8EF4AD69B15D4982559B297BCF1885C529F566660E57EC
68EDBC3C05726CC02FD4CBF4976EAA9AFD5138FE8376435B9FC61D2FC0EB06E3
Generator = 2

[1280 bits]
Modulus (base 16) =
D77946826E811914B39401D56A0A7843A8E7575D738C672A090AB1187D690DC4
3872FC06A7B6A43F3B95BEAEC7DF04B9D242EBDC481111283216CE816E004B78
6C5FCE856780D41837D95AD787A50BBE90BD3A9C98AC0F5FC0DE744B1CDE1891
690894BC1F65E00DE15B4B2AA6D87100C9ECC2527E45EB849DEB14BB2049B163
EA04187FD27C1BD9C7958CD40CE7067A9C024F9B7C5A0B4F5003686161F0605B
Generator = 2

[1536 bits]
Modulus (base 16) =
9DEF3CAFB939277AB1F12A8617A47BBBDBA51DF499AC4C80BEEEA9614B19CC4D
5F4F5F556E27CBDE51C6A94BE4607A291558903BA0D0F84380B655BB9A22E8DC
DF028A7CEC67F0D08134B1C8B97989149B609E0BE3BAB63D47548381DBC5B1FC
764E3F4B53DD9DA1158BFD3E2B9C8CF56EDF019539349627DB2FD53D24B7C486
65772E437D6C7F8CE442734AF7CCB7AE837C264AE3A9BEB87F8A2FE9B8B5292E
5A021FFF5E91479E8CE7A28C2442C6F315180F93499A234DCF76E3FED135F9BB
Generator = 2

[2048 bits]
Modulus (base 16) =
AC6BDB41324A9A9BF166DE5E1389582FAF72B6651987EE07FC3192943DB56050
A37329CBB4A099ED8193E0757767A13DD52312AB4B03310DCD7F48A9DA04FD50
E8083969EDB767B0CF6095179A163AB3661A05FBD5FAAAE82918A9962F0B93B8
55F97993EC975EEAA80D740ADBF4FF747359D041D5C33EA71D281E446B14773B
CA97B43A23FB801676BD207A436C6481F1D2B9078717461A5B9D32E688F87748
544523B524B0D57D5EA77A2775D2ECFA032CFBDBF52FB3786160279004E57AE6
AF874E7303CE53299CCC041C7BC308D82A5698F3A8D0C38271AE35F8E9DBFBB6
94B5C803D89F7AE435DE236D525F54759B65E372FCD68EF20FA7111F9E4AFF73
Generator = 2

In addition to these groups, the following groups MAY be used, each of which has also been rigorously proven to be prime:

[1] The Oakley Group 2 (1024 bit prime) defined in Section E.2 of [RFC2412], pp. 46-47. Primitive roots (acceptable as SRP generators): 5,11,13,19,29,31. Subgroup generators (NOT acceptable): 2,3,7,17,23.

[2] For the 1536-bit [MODP] group, acceptable generators (base 10): 31. NOT acceptable generators: 2,3,5,7,11,13,17,19,23,29.

[3] For the 2048-bit [MODP] group, acceptable generators: 11,13,17,23,29,31. NOT acceptable generators: 2,3,5,7,19.

[4] For the 3072-bit [MODP] group, acceptable generators: 5,7,17,23,31. NOT acceptable generators: 2,3,11,13,19,29.

[5] For the 4096-bit [MODP] group, acceptable generators: 5,13,29,31. NOT acceptable generators: 2,3,7,11,17,19,23.

[6] For the 6144-bit [MODP] group, acceptable generators: 5,11,13,17,23,29. NOT acceptable generators: 2,3,7,19,31.

[7] For the 8192-bit [MODP] group, acceptable generators: 19,23,29,31. NOT acceptable generators: 2,3,5,7,11,13,17.

Appendix B - 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. Other performance data is available in [AESPERF],[PENTPERF] and [UMACPERF].

B.1 Authentication transforms

Table B-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

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 B-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

Table B-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 B-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.

B.2 Encryption and Authentication transforms

Table B-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 B-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 B-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 B-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 B-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.

Data size	AES CBC + CBCMAC	AES CTR + CBCMAC	AES 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 B-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.

Table B-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 B-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, William Dixon of Microsoft, David Black of EMC, Joseph Tardo and Uri Elzur of Broadcom, Julo Satran, Ted Ts'o, Ofer Biran, and Charles Kunzinger of IBM, Allison Mankin of ISI, Mark Bakke and Steve Senum of Cisco, Erik Guttman of Sun Microsystems and Howard Herbert of Intel for useful discussions of this problem space.

Authors' Addresses

Bernard Aboba
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

Phone: +1 425 706 6605
Fax: +1 425 706 7329
EMail: bernarda@microsoft.com

Joshua Tseng
Nishan Systems
3850 **North First Street**
San Jose, CA 95134-1702

Phone: +1 408 519 3749
EMail: jtseng@nishansystems.com

Jesse Walker
Intel Corporation
2211 **NE 25th Avenue**
Hillboro, OR 97124

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

Venkat Rangan
Rhapsody Networks Inc.
3450 **W. Warren Ave.**
Fremont, CA 94538

Phone: +1 510 743 3018
Fax: +1 510 687 0136
EMail: venkat@rhapsodynetworks.com

Franco Travostino
Director, Content Internetworking Lab

Nortel Networks
3 Federal Street
Billerica, MA 01821

Phone: +1 978 288 7708
EMail: travos@nortelnetworks.com

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 implementers 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

Expiration Date

This memo is filed as <draft-ietf-ips-security-14.txt>, and expires February 22, 2003.