To:	INCITS T10 Committee
From:	Matt Ball, Quantum
	David Black, EMC
Date:	3 December 2006
Document:	T10/06-449r0
Subject:	SPC-4: Establishing a Security Association using IKEv2

1 Revision History

Revision 0: Initial version with lots of help from Ralph Weber.
Revision 1: Incorporate comments from discussion in November Las Vegas meeting. Major changes:

- Add timeout support (new STV payload). Timeouts are recommended (should) rather than mandatory (shall).
- Change sequencing support to talk about device server discarding state instead of mandatory timeouts.
- Changed most IKEv2-SCSI-specific ASC/ASCQs to new values.
- Require 16 kilobytes of parameter data support.
- Tweak Certificate Encoding field in Certificate Request payload so that it tells the device server whether or not a URL-based certificate format is acceptable to the application client.
- Make support for skipping authentication optional.
- Specify and explain what not to do with the **PROGRESS INDICATION** sense data.
- Add usage data to SCA payload
- Added the notify (Initial contact only) and delete payloads
- [Added a number of Editor's notes indicating significant additional work to be done ;-).]

2 References

T10/SSC-3r3b, SCSI-3 Stream Commands.
T10/SPC-4r7a, SCSI Primary Commands.
T10/06-369r6 Ralph Weber, Security Association Model for SPC-4.
T10/06-388r3 David Black, SPC-4: Security Goals and Threat Model.
T10/06-225r4 Matt Ball, SSC-3: Key Entry using Encapsulating Security Payload (ESP).
NIST SP 800-56A, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography.
IETF RFC 3280, Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile
IETF RFC 4306, Internet Key Exchange (IKEv2) Protocol.
T11/06-157v3 Fibre Channel - Security Protocols (FC-SP)

3 General

This proposal provides a method, named IKEv2-SCSI, for creating a security association using Diffie-Hellman (DH) key establishment based on IETF RFC 4306 "IKEv2" and guidance from NIST SP 800-56A.

A security association provides the infrastructure necessary for sending encrypted messages between the application client and device server, and allows end-point authentication to prevent man-in-the-middle attacks.

This proposal assumes that 06-369r6 (or later) also passes.

3.1 Differences between IKEv2 and IKEv2-SCSI

The important differences between IKEv2 and IKEv2-SCSI include the following:

a) IKEv2-SCSI has only a single type of SA. An SA created by the IKEv2-SCSI protocol is used to directly protect SCSI traffic. There is no concept of child SAs.
b) The entity sending SCSI traffic determines what SA is used and what is to be protected via appropriate use of the SAI for the SA. SCSI addresses are not involved in this determination, and hence IKEv2-SCSI does not provide address-based data origin authentication; this functionality is left to SCSI transports, as SCSI addresses are transport-specific. SCSI command standards define the uses for SAs and the mechanisms for communicating the applicable SAIs between application clients and device servers.
c) Cryptographic algorithm negotiation has been simplified to reuse a SCSI Device Capabilities design approach. The simplification includes removal of IKEv2's proposal concept; the application client chooses algorithms supported by the device server in accordance with the application client's policy and preferences.
d) Significant portions of IKEv2 have been removed as inapplicable to SCSI. The removed functionality includes Traffic Selectors, NAT Traversal, Remote Configuration, and Compression.

In IKEv2 terminology, the application client is the IKEv2 initiator and the device server is the IKEv2 responder. A device server cannot initiate IKEv2-SCSI.

4 Changes to SPC-4

New additions are in blue.
Editor's notes in purple.

2 Normative references

…

2.4 NIST References

NIST FIPS 186-2, *Digital Signature Standard (DSS)*

2.5 IETF References

RFC 2104, *HMAC: Keyed-Hashing for Message Authentication.*
RFC 2410, *The NULL Encryption Algorithm and Its Use With IPsec.*
RFC 3602, *The AES-CBC Cipher Algorithm and Its Use with IPsec.*
RFC 3526, *More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE).*
RFC 4106, *The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP).*
RFC 4306, *Internet Key Exchange (IKEv2) Protocol.*
RFC 4309, *Using Advanced Encryption Standard (AES) CCM Mode with IPsec Encapsulating Security Payload (ESP).*
RFC 4595, *Use of IKEv2 in the Fibre Channel Security Association Management Protocol.*
RFC 4718, *IKEv2 Clarifications and Implementation Guidelines.*

3.1 Definitions

3.1.a IKEv2-SCSI: Internet Key Exchange protocol version 2 for SCSI (see 5.13.4).
3.1.b IKEv2-SCSI SA establishment transaction: A sequence of SECURITY PROTOCOL commands used to establish a security association between an application client and device server (see 5.13.4).
3.1.c SA Participant: An application client or device server that participates in the creation or use of an SA.

5.13 Security Features

...

Note: See 06-369r6 for 5.13.1 to 5.13.3.

5.13.1.3 Creating a security association

...

Table x2 – Security protocols that create SAs

Security Protocol Code	Description	Reference
zzh	SA Establishment Capabilities	6.27.3
xxh	IKEv2-SCSI	5.13.4

...

5.13.4 Using IKEv2-SCSI to establish a security association

5.13.4.1 Using IKEv2-SCSI to establish a security association overview

The IKEv2-SCSI protocol is a subset of the IKEv2 (see RFC 4306) protocol that is suitable for SCSI in establishing an SA (see 5.13.1).

An IKEv2-SCSI SA establishment transaction occurs between an application client and a device server, and is always initiated by the application client.

The IKEv2-SCSI protocol creates two SAs:

a) An SA that protects data sent from the application client; and
b) An SA that protects data sent from the device server.

IKEv2-SCSI creation of an SA encompasses three phases that shall be performed in the following order:

1. **Device Capabilities**. The application client determines the device server's cryptographic capabilities (see 5.13.4.2);
2. **Key Exchange**. The application client and device server perform a key exchange, determine SAIs, and may complete the creation of the SA (see 5.13.4.3); and
3. **Authentication**. The application client and device server authenticate each other and complete the creation of the SA (see 5.13.4.4).

The values of the **SECURITY PROTOCOL** field and the **SECURITY PROTOCOL SPECIFIC** field in the SECURITY PROTOCOL IN and SECURITY PROTOCOL OUT commands identify the phase for the IKEv2-SCSI protocol (see 7.7.4.1).

The application client requires the results of the Device Capabilities phase to successfully complete the two subsequent phases. An application client is not required to proceed to the Key Exchange phase after the Device Capabilities phase. An application client is not required to perform a Device Capabilities phase immediately prior to an instance of the Key Exchange phase. If the device's capabilities have changed, the Key Exchange phase may not complete successfully, and the Authentication phase will not complete successfully; the application client recovers from these failures by performing the Device Capabilities phase again.

Use of the Authentication phase is negotiated in the other two phases. If both SA participants agree that that the Authentication phase is not used, then the Authentication phase is omitted and SA creation occurs upon the completion of the Key Exchange phase. If both SA participants agree that that the Authentication phase is used, then the application client shall follow the Key Exchange phase with a SECURITY PROTOCOL OUT command that initiates the Authentication phase.

5.13.4.2 Phase 1: Device Capabilities

In the Device Capabilities phase, the application client shall send a SECURITY PROTOCOL IN command with the SECURITY PROTOCOL field set to zzh and the SECURITY PROTOCOL SPECIFIC field in the CDB set to 0101h to read the device server's cryptographic capabilities (see 6.27.3).

The device server uses ALGORITHM TYPE of F9h (see Table K1) in the Device Capabilities phase to report supported IKEv2-SCSI authentication algorithms. The value IKE_AUTH_NONE indicates the absence of IKEv2-SCSI authentication support.

> NOTE – The Device Capabilities phase has no IKEv2 exchange analog in RFC 4306. This phase replaces most of IKEv2's negotiation by having the application client read all of the supported capabilities from the device server.

5.13.4.3 Phase 2: Key Exchange

5.13.4.3.1 Phase 2: Key Exchange overview

The Key Exchange phase consists of an unauthenticated Diffie-Hellman key exchange with nonces (see RFC 4306) and is accomplished in the following two ordered steps:

1) A SECURITY PROTOCOL OUT command with the **SECURITY PROTOCOL** field set to IKEv2-SCSI and the **SECURITY PROTOCOL SPECIFIC** field in the CDB set to 0102h (see 5.13.4.3.2); and
2) A phase 2 SECURITY PROTOCOL IN command with the **SECURITY PROTOCOL** field set to IKEv2-SCSI and the **SECURITY PROTOCOL SPECIFIC** field in the CDB set to 0102h (see 5.13.4.3.3).

5.13.4.3.2 Phase 2: Key Exchange - part 1

The application client shall use the SECURITY PROTOCOL OUT command with the **SECURITY PROTOCOL** field set to IKEv2-SCSI and the **SECURITY PROTOCOL SPECIFIC** field in the CDB set to 0102h to send its key exchange message to the device server. The parameter data shall include the IKE header (HDR) and four payloads, in order:

1) STVi (see 7.7.4.12);
2) SCAi (see 7.7.4.11);
3) KEi (see 7.7.4.3); and
4) NONCEi (see 7.7.4.7).

The STVi payload shall contain the inactivity timeouts that apply to this instance of IKEv2-SCSI and the SA that is created.

The SCAi Payload shall contain the cryptographic algorithms selected by the application client and the intended usage of the created SAs.

The KEi Payload shall contain the application client's Diffie-Hellman value.

The NONCEi Payload shall contain the application client's random nonce.

If the device server receives an SCAi payload containing an unsupported algorithm or key length, then the device server shall terminate the command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST and the additional sense code set to IKEv2-SCSI PARAMETER VALUE INVALID. The sense data shall contain a sense key specific sense data descriptor for the ILLEGAL REQUEST sense key in which the **FIELD POINTER** field designates the position of the first unsupported algorithm or key length.

If the device server receives an SCAi payload that does not contain all the required transforms, then the device server shall terminate the command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST and the additional sense code set to PARAMETER LIST INCOMPLETE.

[Editor's Note: PARAMETER LIST INCOMPLETE is a new ASC/ASCQ - suggest 26h/13h]

5.13.4.3.3 Phase 2: Key Exchange - part 2

If the device server successfully completes Phase 2: Key Exchange - part 1, then the application client shall send a SECURITY PROTOCOL IN command with the **SECURITY PROTOCOL** field set to IKEv2-SCSI IN and with the **SECURITY PROTOCOL SPECIFIC** field in the CDB set to 0102h to obtain the device server's key exchange message. The parameter data returned by the device server shall contain the IKE header (HDR), and the following payloads, in order:

1) SCAr (see 7.7.4.12);
2) KEr (see 7.7.4.11);
3) NONCEr (see 7.7.4.7); and
4) Zero or more CERTREQ payloads (see 7.7.4.5).

While processing Phase 2: Key Exchange - part 2, the device server shall:

a) associate this to the previous SECURITY PROTOCOL IN command according to the I_T_L nexus;
b) echo the cryptographic algorithms supplied by the application client;
c) provide its SAI in the SCAr Payload;
d) complete the Diffie-Hellman exchange with the KEr Payload; and
e) send its nonce in the NONCEr Payload.

The device server may use the optional CERTREQ payload(s) to specify its trust anchors list when PKI-based Authentication is being used (see RFC 3280).

> NOTE – The KE*, NONCE* and CERTREQ payloads are identical to those used in IKEv2 (see RFC 4306). The SCA* payloads are simplified from their IKEv2 counterparts (SA* payloads) because device server capabilities are determined in phase 1, and because the SAIs carried in the IKE header do not need to be repeated in this payload.

If the device server receives a SECURITY PROTOCOL IN command with the **SECURITY PROTOCOL** field set to IKEv2-SCSI and the **SECURITY PROTOCOL SPECIFIC** field in the CDB set to 0102h and the device server has not successfully completed a phase 2 IKEv2-SCSI SECURITY PROTOCOL OUT command on the same I_T_L nexus, then the SECURITY PROTOCOL IN command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST and the additional sense code set to IKEv2-SCSI INVALID COMMAND SEQUENCE.

[Editor's Note: IKEv2-SCSI INVALID COMMAND SEQUENCE is a new ASC/ASCQ - suggest 2Ch/10h]

5.13.4.3.3 Phase 2: Key Exchange - phase completion

After successfully transferring both messages in Phase 2: Key Exchange, the SA participants shall:

a) generate SKEYSEED (see RFC 4306) using the specified pseudo-random function before proceeding to the Authentication phase, if applicable; and then
b) use SKEYSEED to generate the following keys: SK_d, SK_ai, SK_ar, SK_ei, SK_er, SK_pi, and SK_pr (see RFC 4306).

The following keys constitute KEYMAT (see Table x1 of 06-369r6) and are derived from SKEYSEED (see RFC 4306):

a) SK_d: A key for deriving further keys for use with these SAs. This is recorded as KEY_SEED in the resulting SCSI security association (See TBD).
b) SK_ai and SK_ar: IKEv2 authentication keys for use by the application client (SK_ai) and the device server. IKEv2 refers to these as authentication keys, but their function is to provide cryptographic integrity protection for subsequent IKEv2 messages.
c) SK_ei and SK_er: IKEv2 encryption keys for use by the application client (SK_ei) and the device server to protect subsequent IKEv2 messages.

d) SK_pi and SK_pr: IKEv2 pseudo-random function keys that participate in the generation of the AUTH payloads. These keys cryptographically bind the authenticated identities to this cryptographic exchange.

If the application client selects the IKE_AUTH_NONE value for algorithm type F9h in the Key Exchange phase, and phase 2 completes without errors, the application client shall not perform the Authentication phase (phase 3) and shall only generate SK_d.

> NOTE – If IKEv2-SCSI authentication is not performed, no security is provided against man-in-the-middle attacks (see RFC 3552). In this situation an adversary that can interpose itself between the application client and device server may obtain full access (read and modify) to communications protected by SAs created by IKEv2-SCSI without the knowledge of the application client or device server. It is not acceptable to omit IKEv2-SCSI authentication unless man-in-the-middle attacks are not of concern or are prevented by other means such as physical security of the transport (e.g., a direct physical connection) assurance that the service delivery subsystem does not permit man-in-the-middle (e.g., the subsystem is contained in a single equipment rack and has a single switch whose configuration and management are tightly controlled), or transport security measures that prevent man-in-the-middle attacks (e.g., use of IPsec with iSCSI). Deliberate administrative action is required to omit authentication; 7.7.4.11.5 requires that omitting authentication not be a default for device servers and application clients.

> NOTE – The ability to omit authentication in IKEv2-SCSI has no protection against a downgrade attack because modified communications have the ability to cause two parties that would otherwise authenticate to not authenticate. The decision by a device server to offer the capability of omitting authentication in the Device Capabilities phase and the decision by an application client to select omission of authentication in the Key Exchange phase are security policy decisions that absence of authentication is acceptable. When absence of authentication is not acceptable, a device server should not offer the capability of omitting authentication in the Device Capabilities phase and/or an application client should not select the capability of omitting authentication in the Key Exchange phase.

> NOTE – The key exchange phase corresponds to the IKEv2 IKE_SA_INIT exchange in RFC 4306, except that determination of device server capabilities has been moved into phase 1.

5.13.4.4 Phase 3: Authentication

5.13.4.4.1 Phase 3: Authentication - general

The Authentication phase performs the following functions:

a) authenticates both the application client and the device server;
b) protects the previous phases of the protocol; and
c) cryptographically binds the authentication and previous phases to the generated SA.

The Authentication phase is accomplished in the following two steps:

1. A SECURITY PROTOCOL OUT command with the SECURITY PROTOCOL field set to IKEv2-SCSI (i.e., xxh) and the SECURITY PROTOCOL SPECIFIC field in the CDB set to 0103h (see 5.13.4.4.2); and
2. A phase 2 SECURITY PROTOCOL IN command with the SECURITY PROTOCOL field set to IKEv2-SCSI and the SECURITY PROTOCOL SPECIFIC field in the CDB set to 0103h (see 5.13.4.4.3).

The parameter data for both commands shall be encrypted and integrity protected by the algorithms and keys determined in the Key Exchange phase.

If the device server receives a SECURITY PROTOCOL IN command with the SECURITY PROTOCOL field set to IKEv2-SCSI and the SECURITY PROTOCOL SPECIFIC field in the CDB set to 0103h and the device server does not have state from the successful completion of a phase 3 IKEv2-SCSI SECURITY

PROTOCOL OUT command on the same I_T_L nexus, the SECURITY PROTOCOL IN command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST and the additional sense code set to IKEv2-SCSI INVALID COMMAND SEQUENCE.

If the device server receives a SECURITY PROTOCOL OUT command with the SECURITY PROTOCOL field set to IKEv2-SCSI and the SECURITY PROTOCOL SPECIFIC field in the CDB set to 0103h and the device server does not have state from the successful completion of IKEv2-SCSI phase 2 on the same I_T_L nexus, the SECURITY PROTOCOL OUT command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST and the additional sense code set to IKEv2-SCSI INVALID COMMAND SEQUENCE.

5.13.4.4.2 Phase 3: Authentication - part 1

The application client shall send a SECURITY PROTOCOL OUT command with the SECURITY PROTOCOL field set to IKEv2-SCSI and the SECURITY PROTOCOL SPECIFIC field in the CDB set to 0103h to send its authentication message to the device server. This message consists of the IKE header (HDR), plus an encrypted payload, E, that shall contain the IDi payload, one or more optional CERT payload(s), one or more optional CERTREQ payload(s), an optional N payload, and the AUTHi payload in that order.

The application client shall:

a) assert its identity with the IDi Payload;
b) prove knowledge of the secret corresponding to IDi; and
c) integrity protect the prior phases using the AUTHi Authentication Payload (see x.x.x).

The application client may send its Certificate(s) in CERT Payload(s) and may independently send a list of its trust anchors in CERTREQ Payload(s) (see RFC 4306). If any CERT Payloads are included, the first Certificate provided shall contain the public key used to verify the Authentication Payload. The CERT and CERTREQ payloads are independently optional because the application client and device server may use different authentication methods.

The application client uses the Notify (N) payload to carry an Initial Contact notification. The Initial Contact notification informs the device server that this newly created IKEv2-SCSI SA will be the only SA between the device server and this application client. The device server may use this information to delete all other SAs with the same application client (as indicated by identical IDi), but shall do so only after completion of phase 3. In particular, a device server shall ignore an N-IC payload if there is a failure in authentication of the application client.

The device server shall check the Authentication in the AUTHi payload. The CERT payload(s) are used as part of this for PKI-based authentication. If the authentication fails, or there are other reasons why the SA setup is unable to proceed, the SECURITY PROTOCOL OUT command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST and the additional sense code set to AUTHENTICATION FAILED.

5.13.4.4.3 Phase 3: Authentication - part 2

The application client then sends a SECURITY PROTOCOL IN command with the SECURITY PROTOCOL field set to IKEv2-SCSI and the SECURITY PROTOCOL SPECIFIC field in the CDB set to 0103h to obtain the device server's authentication message. This message consists of the IKE header (HDR), plus an encrypted payload, E, that shall contain the IDr payload, one or more optional CERT payload(s) and the AUTHr payload in that order.

The device server associates this to the previous SECURITY PROTOCOL OUT command by virtue of it being for the same security protocol on the same I_T_L nexus.

The device server:

a) asserts its identity with the IDr Payload, optionally sends one or more Certificates, with the first Certificate containing the public key used to verify the Authentication Payload listed first;
b) authenticates its identity; and
c) protects the integrity of the prior phase messages with the Authentication Payload.

> NOTE – The Authentication phase corresponds to the IKEv2 IKE_AUTH exchange in RFC 4306.

5.13.5 Security progress indication

The cryptographic calculations required by some security protocols can consume a significant amount of time in the device server. If the device server receives a SECURITY PROTOCOL OUT command or SECURITY PROTOCOL IN command that it is unable to process because required calculations are not complete, the command shall be terminated with CHECK CONDITION status, with the sense key set to NOT READY and the additional sense code set to LOGICAL UNIT NOT READY, OPERATION IN PROGRESS. The sense data should include sense key specific data for the NOT READY sense key that contains a **PROGRESS INDICATION** field indicating the progress of the device server in performing the necessary calculations.

The device server shall not use the **PROGRESS INDICATION** to report the actual progress of any cryptographic computation that may take a variable amount of time based on its inputs. The device server may use the **PROGRESS INDICATION** to report synthetic progress that does not reveal the actual progress of the computation prior to its completion (e.g., divide a constant expected time for the computation by 10 and advance the **PROGRESS INDICATION** by 10% increments based solely on the time).

> NOTE – This restriction applies to implementations of Diffie-Hellman computations and operations involving public or asymmetric keys (e.g., RSA) that optimize operations on large numbers based on the values of inputs (e.g., a computational step may be skipped when a bit or set of bits in an input is zero). A **PROGRESS INDICATION** that advances based on the computation structure (e.g., count of computational steps) may reveal the time taken by content-dependent portions of the computation, thereby revealing information about its inputs. Many implementations of symmetric encryption (e.g., AES) and secure hashes (e.g., SHA-1) take a constant amount of time independent of their inputs, and hence are not subject to this restriction.

When cryptographic calculations are in progress, the sense data specified in the first paragraph of this subsection shall be returned in response to a REQUEST SENSE command. Upon completion of the calculations, the sense data returned for a REQUEST SENSE command shall have the sense key set to NO SENSE.

6.27 SECURITY PROTOCOL IN command

Editor's Note: Add the following code to Table 173 "SECURITY PROTOCOL field in SECURITY PROTOCOL IN command" from the Reserved area:

Code	Description	Reference
zzh	**SA Establishment Capabilities**	6.27.3
xxh	IKEv2-SCSI IN	6.27.4

Editor's note: The values zzh and xxh should be consecutive, if possible, and are both chosen by the CAP working group.

...

6.27.3 SA Establishment Capabilities description

The SA Establishment Capabilities security protocol returns information about the SA establishment protocols that are supported by the device server. The information returned depends on the value in the SECURITY PROTOCOL SPECIFIC field in the SECURITY PROTOCOL IN command CDB (see 7.7.2).

If any SA establishment security protocols are supported, the SA Establishment Capabilities security protocol shall be supported.

6.27.4 IKEv2-SCSI SECURITY PROTOCOL IN description

6.27.4.1 IKEv2-SCSI SECURITY PROTOCOL IN protocol specific

Table A1 describes the **SECURITY PROTOCOL SPECIFIC** codes for IKEv2-SCSI IN protocol of the SECURITY PROTOCOL IN command. These codes indicate the phase of the protocol.

Table A1 – SECURITY PROTOCOL SPECIFIC field for IKEv2-SCSI IN protocol

Code	Description	Support	Reference
0000h – 00FFh	Restricted		RFC 4306
0100h – 0101h	Reserved		
0102h	Key Exchange phase	Mandatory	6.27.4.3
0103h	Authentication phase	Mandatory	6.27.4.3
0104h	Reserved [Editor's Note: for Delete]	Mandatory	
0105h – EFFFh	Reserved		
F000h – FFFFh	Vendor Specific		

6.27.4.3 IKEv2-SCSI for the SECURITY PROTOCOL IN command

See 7.7.3 for a description of the parameter data format for the IKEv2-SCSI security protocol in the SECURITY PROTOCOL IN command.

If a device server supports IKEv2-SCSI IN for SECURITY PROTOCOL IN, then the device server shall support IKEv2-SCSI OUT for SECURITY PROTOCOL OUT (see 6.28.2.2) and IKEv2-SCSI SUPPORT for SECURITY PROTOCOL IN (see 6.27.4).

6.27.4.4 IKEv2 protocol details and variations for IKEv2-SCSI

The IKEv2 protocol details and variations specified in RFC 4306 apply to IKEv2-SCSI as follows:

a) A parameter data size of at least 16 kilobytes shall be supported for the SECURITY PROTOCOL IN and SECURITY PROTOCOL OUT commands.

b) Retransmission is the responsibility of the SCSI transport; the timeout and retransmission mechanisms in RFC 4306 shall not be used. Each SCSI command used by IKEv2-SCSI completes by conveying a status response from the device server.

c) IKEv2-SCSI uses the Message ID field for sequencing (see RFC 4306), but only for SA setup (see 7.7.3.3).

d) Overlapping IKEv2-SCSI requests (see RFC 4306) are prohibited on an I_T_L nexus. If an application client attempts to overlap requests, the offending command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB or PARAMETER VALUE INVALID (see 5.13.4 and 7.7.3.5).

e) [Editor's Note: State Synchronization (Section 2.4 of RFC 4306) will apply in principle. Checking whether an SA still exists at the device server will be specified in a future revision of this proposal that specifies the Delete payload for IKEv2-SCSI.]
f) IKEv2 version numbers (see RFC 4306) are used in IKEv2-SCSI (see 7.7.3.3), but the ability to respond to an unsupported version number with the highest version number that should be used is not supported, and IKEv2-SCSI does not check for version downgrade.
g) IKEv2 cookies (see RFC 4306) are not used in IKEv2-SCSI.
h) IKEv2 cryptographic algorithm negotiation (see RFC 4306) is replaced by a different negotiation framework in IKEv2-SCSI (see 5.13.4 and 7.7.4.11). The IKEv2 proposal construct is not used in IKEv2-SCSI.
i) An IKEv2-SCSI SA is rekeyed by replacing it with a new SA. CHILD_SAs are not used in IKEv2-SCSI. The RFC 4306 discussion of CHILD_SAs does not apply to IKEv2-SCSI. Coexistence of the original SA and the new SA that is created for rekeying purposes should be supported. IKEv2 does not support rekeying notification for IKE_SAs, therefore IKEv2-SCSI does not support rekeying notification
j) Traffic Selectors (see RFC 4306) are not used by IKEv2-SCSI.
k) The requirements in RFC 4306 on Nonces shall be followed for nonces used in IKEv2-SCSI.
l) The RFC 4306 requirements on Address and Port Agility is specific to the UDP and IP protocols and does not apply to IKEv2-SCSI.
m) Diffie-Hellman exponential reuse and reuse of analogous Diffie-Hellman public values for Diffie-Hellman mechanisms not based on exponentiation is permitted in IKEv2-SCSI as specified in RFC 4306. Freshness and randomness of the nonces are critical to the security of IKEv2-SCSI when Diffie-Hellman exponentials and public values are reused.
n) Keys for the Authentication phase of IKEv2-SCSI shall be generated as specified in RFC 4306.
o) IKEv2-SCSI uses a slightly modified version of the Authentication calculations in RFC 4306 (see 7.7.4.6).
p) RFC 4306 sections on Extensible Authentication Protocol Methods, Generating Keying Material for CHILD_SAs, Rekeying an IKE_SA using CREATE_CHILD_SA, Requesting an Internal Address, Requesting the Peer's Version, IPComp, NAT Traversal, and Explicit Congestion Notification describe mechanisms that are not used in IKEv2-SCSI.
q) IKEv2 Error Handling (see RFC 4306) is replaced by the use of CHECK CONDITION in IKEv2-SCSI (see 5.13.4 and 7.7.3.5).

6.28 SECURITY PROTOCOL OUT command

6.28.1 SECURITY PROTOCOL OUT command description

Editor's Note: Add the following code to Table 178 — "SECURITY PROTOCOL field in SECURITY PROTOCOL OUT command" from the Reserved area:

Code	Description	Reference
xxh	IKEv2-SCSI OUT	6.28.2

Editor's note: Both of the xxh codes mentioned in Tables 173 and 178 should be the same - xxh is the **second** code in table 173.

6.28.2 IKEv2-SCSI OUT description

6.28.2.1 IKEv2-SCSI OUT protocol specific

Table D1 describes the valid SECURITY PROTOCOL SPECIFIC codes for IKEv2-SCSI OUT security protocol of the SECURITY PROTOCOL OUT command.

Table D1 – SECURITY PROTOCOL SPECIFIC field for IKEv2-SCSI OUT protocol

Code	Description	Support	Reference

Code	Description	Support	Reference
0000h – 00FFh	Restricted		RFC 4306
0100h – 0101h	Reserved		
0102h	Key Exchange phase	Mandatory	6.28.2.2
0103h	Authentication phase	Optional	6.28.2.2
0022h – EFFFh	Reserved		
F000h – FFFFh	Vendor Specific		

6.28.2.2 IKEv2-SCSI for SECURITY PROTOCOL OUT

See 7.7.3 for a description of the format of the parameter data for the IKEv2-SCSI security protocol in the SECURITY PROTOCOL OUT command.

If a device server supports the IKEv2-SCSI security protocol in the SECURITY PROTOCOL OUT command then the device server shall also support the IKEv2-SCSI protocol in the SECURITY PROTOCOL IN command (see 6.27.3).

...

7.7 Security protocol parameters

7.7.1 Security protocol information parameters

[Editors Note: Move the contents of SPC-4 r07 6.27.2 to here]

7.7.2 SA Establishment Capabilities parameters

7.7.2.1 Overview

If the **SECURITY PROTOCOL**field is set to SA Establishment Capabilities (i.e., zzh), the contents of the SECURITY PROTOCOL IN the **SECURITY PROTOCOL SPECIFIC** field specifies the parameter data format returned by the command (see table A0).

Table A0 – SECURITY PROTOCOL SPECIFIC field for the SA Establishment Capabilities protocol

Code	Description	Support	Reference
0000h – 0100h	Reserved		
0101h	IKEv2-SCSI Device Capabilities phase	Mandatory	7.7.2.2
0102h – EFFFh	Reserved		
F000h – FFFFh	Vendor Specific		

[Editors note: It is intended that the definition of a second SA Establishment security protocol be accompanied by the addition of a Supported SA Establishment protocols code (probably 0000h or 0100h) and that the support requirement for the IKEv2-SCSI be changed from Mandatory to Optional at that time. However, it is also possible to introduce new SA Establishment protocols by modifying the data in the SCSI Cryptographic Algorithms IKE Payload (see 7.7.4.11).]

7.7.2.2 IKEv2-SCSI Device Capabilities phase parameter data

The IKEv2-SCSI Device Capabilities parameter data indicates the IKEv2 transforms (i.e., key exchange and authentication protocols) supported by the device server (see table ROW1).

Table ROW1 – IKEv2-SCSI Device Capabilities parameter data

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)	PARAMETER DATA LENGTH (n-3)						
3								(LSB)
IKE payload								
4	IKE SCSI Cryptographic Algorithms Payload (see 7.7.4.11)							
n								

The PARAMETER DATA LENGTH field indicates the number of bytes that follow in the parameter data.

The IKE Payload for SCSI Cryptographic Algorithms (see 7.7.4.11) indicates the algorithms supported by the Key Exchange phase (see 5.13.4.3) and Authentication phase (see 5.13.4.4).

The NEXT PAYLOAD field shall be set to zero in the IKE Payload for SCSI Cryptographic Algorithms.

7.7.3 IKEv2-SCSI parameters

7.7.3.1 IKEv2-SCSI parameters overview

Table E1 shows the parameter data format used by a SECURITY PROTOCOL IN command or a SECURITY PROTOCOL OUT command with a SECURITY PROTOCOL field set to IKEv2-SCSI (i.e., xxh). See RFC 4306 for a detailed explanation of the IKE header.

Table E1 – IKEv2-SCSI parameter data

Bit Byte	7	6	5	4	3	2	1	0
IKE HEADER								
0	(MSB)	IKE_SA APPLICATION CLIENT SAI						
7								(LSB)
8	(MSB)	IKE_SA DEVICE SERVER SAI						
15								(LSB)
16	NEXT PAYLOAD							
17	MAJOR VERSION (2h)				MINOR VERSION (0h)			
18	PHASE NUMBER							
19	RESERVED			INTTR	VERSION	RSPNS	RESERVED	
20	(MSB)	MESSAGE ID						
23								(LSB)
24	(MSB)	LENGTH (n+1)						
27								(LSB)
IKE PAYLOADS								
28	IKE PAYLOADS (VARIABLE)							
n								

The IKE_SA APPLICATION CLIENT SAI field contains a value chosen by the application client to uniquely identify its representation of the security association that is being negotiated. This field shall not be set to zero.

The IKE_SA DEVICE SERVER SAI field contains a value chosen by the device server to uniquely identify itself within the context of the security association that is being negotiated. This field:

a) shall be set to zero for the the Key Exchange phase SECURITY PROTOCOL OUT command sent from the application client to the device server; and

b) shall not be zero for any subsequent parameter data.

The NEXT PAYLOAD field contains an identifier that describes the first payload within the IKE PAYLOADS (VARIABLE) field (see 7.7.4.1).

The MAJOR VERSION field shall contain the value 2h. If the device server receives an IKE header with a MAJOR VERSION field containing any other value, the device server shall terminate the command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST and the additional sense code set to PARAMETER VALUE INVALID.

The MINOR VERSION field shall be set to 0h and ignored upon receipt.

The PHASE NUMBER field shall be set to the phase of the IKEv2-SCSI protocol:

a) 2h for the Key Exchange phase (see 5.13.4.3); or

b) 3h for the Authentication phase (see 5.13.4.4).

The initiator (INTTR) bit shall be set to one for SECURITY PROTOCOL OUT commands and shall be set to zero for SECURITY PROTOCOL IN commands. The recipient shall not process a message with the wrong value in the INTTR bit. If the device server receives an IKE header with the INTTR bit set to zero, then the device server shall terminate the command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST and the additional sense code set to PARAMETER VALUE INVALID.

The VERSION bit shall be set to one and ignored upon receipt.

A response (RSPNS) bit set to one indicates that this parameter data is in response to a previous command with the same MESSAGE ID. A RSPNS bit set to zero indicates that this is parameter data is not associated with any previous MESSAGE ID of the same value.

The MESSAGE ID field contains an incrementing value that identifies a particular message and response pair. The first MESSAGE ID in the Key Exchange phase shall be zero. The application client shall increment the MESSAGE ID for each subsequent message. The device server shall respond with the same MESSAGE ID that the application client used in the initial command. Neither the application client nor the device server shall process an IKEv2-SCSI payload that contains a lower MESSAGE ID than the largest one previously seen (see RFC 4306).

The LENGTH field shall contain the total number of bytes to be transferred for this IKEv2-SCSI message, including the header and all the IKE payloads.

The IKE PAYLOADS (VARIABLE) field contains one or more IKE payloads (see table F1).

Table F1 – IKE payload format

Bit Byte	7	6	5	4	3	2	1	0
0	NEXT PAYLOAD							
1	CRIT	RESERVED						
2	(MSB)	PAYLOAD LENGTH (m+1)						
3								(LSB)
IKE PAYLOAD DATA								

Bit Byte	7	6	5	4	3	2	1	0
...	PAYLOAD DATA (VARIABLE)							
m								

The NEXT PAYLOAD field contains a value from Table G1 (see 7.7.4.1) that describes the payload that follows the current payload, if any. The current payload is described by the preceding NEXT PAYLOAD field either in the IKE header (see Table E1) or in the preceding payload (see Table F1). The last payload occurs when either the NEXT PAYLOAD field is set to 00h (No Next Payload) or the current payload is of type 2Eh (Encrypted). If the current payload is encrypted (i.e. 2Eh), then the NEXT PAYLOAD field identifies the first encrypted payload.

A critical (CRIT) bit set to one indicates that the sender does not want the receiver to ignore the payload. A CRIT bit set to zero indicates that the receiver shall ignore any payloads that the receiver does not recognize. The CRIT bit shall be set to one in all payloads defined in this standard except the Vendor ID payload (see RFC 4306).

The PAYLOAD LENGTH field contains the length of the entire payload, including the payload header and the payload data.

The PAYLOAD DATA (VARIABLE) field contains a payload (see 7.7.4).

7.7.4 IKE Payloads

7.7.4.1 Overview

Table G1 shows the possible values of the NEXT PAYLOAD field. In Table G1, the column entitled “IN Support” indicates the required support level of the device server for a particular payload value for the SECURITY PROTOCOL IN command. The column entitled “OUT Support” indicates the required support level of the device server for a particular payload value for the SECURITY PROTOCOL OUT command.

Table G1 – Values for NEXT PAYLOAD

Value	Notation	Description	IN Support	OUT Support	Reference
00h		No Next Payload	Mandatory	Mandatory	7.7.4.2
01h-20h		Reserved			
21h	SA	Security Association [a]	Prohibited	Prohibited	RFC 4306
22h	KE	Key Exchange	Mandatory	Mandatory	7.7.4.3
23h	IDi	Identification – Application Client	Prohibited	Mandatory	7.7.4.4
24h	IDr	Identification – Device Server	Mandatory	Prohibited	7.7.4.4
25h	CERT	Certificate	Optional	Optional	7.7.4.5
26h	CERTREQ	Certificate Request	Optional	Optional	7.7.4.5
27h	AUTH	Authentication	Mandatory	Mandatory	7.7.4.6
28h	NONCE	Nonce	Mandatory	Mandatory	7.7.4.7
29h	N [b]	Notify	None	Optional	7.7.4.8a

2Ah	D	Delete	None	Mandatory	7.7.4.8[b]
2Bh	V	Vendor ID	Mandatory	Mandatory	7.7.4.9
2Ch	TSi	Traffic Selector – Application Client	Prohibited	Prohibited	RFC 4306
2Dh	TSr	Traffic Selector – Device Server	Prohibited	Prohibited	RFC 4306
2Eh	E	Encrypted	Mandatory	Mandatory	7.7.4.10
2Fh	CP	Configuration	Prohibited	Prohibited	RFC 4306
30h	EAP	Extensible Authentication	Prohibited	Prohibited	RFC 4306
31h-7Fh		Restricted			RFC 4306
80h		Reserved			
81h	SCA	SCSI Cryptographic Algorithms	Mandatory	Mandatory	7.7.4.11
82h	STV	SCSI Timeout Values	Mandatory	Mandatory	7.7.4.12
83h-BFh		Reserved			
C0h-FFh		Vendor Specific			

[a] This payload type value is not used in IKEv2-SCSI. The SCSI Cryptographic Algorithms payload (i.e., 81h) is used instead.

[b] The Notify payload is used only to carry an Initial Contact notification. All other notifications defined in RFC 4306 are Prohibited.

7.7.4.2 No Next Payload

An IKE payload type value of 00h indicates that there is no following payload.

7.7.4.3 Key Exchange payload

An IKE payload type of 22h indicates that this payload contains key exchange data. Table H1 shows the format of this key exchange data.

Table H1 – IKE Key Exchange payload format

Bit Byte	7	6	5	4	3	2	1	0
0	NEXT PAYLOAD							
1	CRIT	RESERVED						
2	(MSB)	PAYLOAD LENGTH (m+1)						
3								(LSB)
IKE KEY EXCHANGE PAYLOAD DATA								
4	DIFFIE-HELLMAN GROUP NUMBER							
5								
6	RESERVED							
7	RESERVED							
8	KEY EXCHANGE DATA							

Bit Byte	7	6	5	4	3	2	1	0
m								

The NEXT PAYLOAD field, CRIT bit, and PAYLOAD LENGTH field are defined in 7.7.4.1.

The DIFFIE-HELLMAN GROUP NUMBER field contains a value that identifies the Diffie-Hellman group being used for this key exchange (see 7.7.4.11.4).

The KEY EXCHANGE DATA field contains the sender's Diffie-Hellman public value for this key exchange. The format of KEY EXCHANGE DATA is as specified in the reference cited in that registry for the value used. When a prime modulus (mod p) Diffie-Hellman group is used, the length of the Diffie-Hellman public value shall be equal to the length of the prime modulus over which the exponentiation was performed; zero bits shall be prepended to the value if necessary. Diffie-Hellman exponential reuse and reuse of the analogous Diffie-Hellman public values for Diffie-Hellman mechanisms not based on exponentiation is permitted as specified in RFC 4306.

7.7.4.4 Identification payload

IKE payload type values of 23h and 24h indicate Identification payloads, for the application client (i.e., initiator) and device server (i.e., responder), respectively. The format is identical to the IKEv2 payload format in RFC 4306. The ID Type shall be one of the following:

a) ID_DER_ASN1_DN and ID_DER_ASN1_GN may be used when the sender of this payload will present a certificate to authenticate its identity. They shall not be used when certificates are not used; or
b) ID_KEY_ID allows arbitrary identity data to be passed. SCSI port and device names may be passed using this type.

Other ID Types shall not be used.

If the device server receives any other ID Type, then the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST and the additional sense code set to IKEv2-SCSI PARAMETER VALUE INVALID.

7.7.4.5 Certificate and Certificate Request payloads

An IKE payload type value of 25h indicates a Certificate payload, and an IKE payload type value of 26h indicates a Certificate Request payload conveying a preferred trust anchor as part of a certificate request (see RFC 4306) Certificate formats shall be as defined in RFC 3280 or FC-SP.

One of the functions of the Certificate Request payload is to inform the device server whether the application client supports URL-based certificate formats. If the CERTIFICATE ENCODING field in the Certificate Request payload contains a URL-based format, the device server may use that format or may send the actual certificate or certificates. If the CERTIFICATE ENCODING field in the Certificate Request payload does not contain a URL-based format, the device server shall not use a URL-based certificate format.

> NOTE: This support for URL-based certificate formats is restricted by comparison to that specified for IKEv2 in RFC 4306. RFC 4306 permits an entity to say that it supports URL-based certificate formats, but nonetheless request a non-URL based format. SCSI does not have the maximum payload size issues that cause IKEv2 certificate difficulties, and hence IKEv2-SCSI requires the application client to ask for a URL-based format if it wants the device server to use one.

[Editor's Note: Need to figure out whether that NOTE is acceptable, as the alternatives of a) using separate authentication algorithm values for URL vs. non-URL certificates, 2) adding indication of HTTP support to the Certificate Request payload and 3) supporting the Notify payload are all unpleasant IKEv2 changes and additions of complexity. Everything is simpler if the device server defaults to just having the certificate.]

[Editor's Note: Will need to restrict certificate encodings. See RFC 4718 for a starting point. FC-SP insists on using Base-64 to transmit certificates. DLB is inclined to "Just Say No" to that.]

[Editor's Note: Need to check identity type support against RFC 3280, and in particular figure out how RFC 3280's SubjectAltName support shows up in the Identity field. We may need a new identity type for the FC-SP certificates.]

7.7.4.6 Authentication payload

An IKE payload type of 27h indicates an Authentication payload. The payload format is based on that specified in RFC 4306 with the field structure unchanged. The computation of the AUTHENTICATION DATA field is based on the algorithm specified in RFC 4306, with the following changes and clarifications for SCSI:

a) A shared key used to calculate a Shared Key Message Integrity Code (i.e., Auth Method 2) shall be associated with one identity. The same pre-shared key shall not be used to authenticate both an application client and a device server. Use of the same pre-shared key for a group of application clients or a group of device servers is strongly discouraged, as it enables any member of the group to impersonate any other member.
b) RSA and DSS Digital Signature support is optional. Shared Key authentication shall be supported.
c) The device server prepends the contents of its supported IKE cryptographic mechanisms page to its Key Exchange phase IKEv2-SCSI message in constructing the block of data to be signed.
d) The shared key signing mechanism shall use the 22 ASCII character pad string "Key Pad for IKEv2-SCSI" without null termination in place of the 17 ASCII character pad string "Key Pad for IKEv2" (see RFC 4306).

7.7.4.7 Nonce payload

An IKE payload type value of 28h indicates a Nonce payload that carries a random nonce. Randomness of nonces is crucial to the security of IKEv2. See RFC 4306 for the specification of the Nonce payload.

7.7.4.8a Notify payload

An IKE payload type value of 29h indicates a Notify payload. IKEv2-SCSI uses the Notify payload solely for Initial Contact Support.

The Initial Contact notification informs the device server that the SA pair established by this IKEv2-SCSI instance is the only SA between the device server and this application client, as identified by IDi (see 7.7.4.4). After successful completion of this IKEv2-SCSI instance, the device server may delete other SAs to the same application client without waiting for the appropriate timeouts. The device server shall not act upon an Initial Contact notification if application client authentication fails.

Table H2 shows the format of the Notify payload.

Table H2 – Notify payload format

Bit Byte	7	6	5	4	3	2	1	0
0	NEXT PAYLOAD							

Bit Byte	7	6	5	4	3	2	1	0
1	CRIT	Reserved						
2	(MSB)	PAYLOAD LENGTH (16)						
3								(LSB)
Initial contact notification data								
4	PROTOCOL ID (1)							
5	SAI SIZE (8)							
6	NOTIFY MESSAGE TYPE (16384)							
7								
8	SAI							
15								

The NEXT PAYLOAD field, CRIT bit, and PAYLOAD LENGTH field are defined in 7.7.4.1.

The PROTOCOL ID field shall be set to 1h to indicate IKEv2-SCSI SAs.

The SAI SIZE field shall be set to 8h.

The NOTIFY MESSAGE TYPE field shall be set to 16384 to indicate an Initial Contact notification (this value is defined in RFC 4306).

The SAI field shall be set to the device server's SAI for this IKEv2-SCSI instance.

7.7.4.8b Delete payload

An IKE payload type value of 2Ah indicates a Delete payload. The device server shall only process a Delete payload if it is contained within an Encrypted payload that has a valid ICV. Table H3 shows the format of a Delete payload.

Table H3 – Delete payload format

Bit Byte	7	6	5	4	3	2	1	0
0	NEXT PAYLOAD							
1	CRIT	Reserved						
2	(MSB)	PAYLOAD LENGTH (n)						
3								(LSB)
Initial contact notification data								
4	PROTOCOL ID							
5	SAI SIZE							
6	NUMBER OF SAIs (0001h)							
7								
8	(MSB)	APPLICATION CLIENT SECURITY ASSOCIATION INDEX						

Bit Byte	7	6	5	4	3	2	1	0
15								(LSB)

The NEXT PAYLOAD field, CRIT bit, and PAYLOAD LENGTH field are defined in 7.7.4.1.

The PROTOCOL ID field shall be set to 01h to indicate IKEv2-SCSI SAs.

The SAI SIZE field shall be set to 08h.

The NUMBER OF SAIs field shall be set to 0001h.

The APPLICATION CLIENT SECURITY ASSOCIATION INDEX field contains the SAI of a security association that the application client has removed from its internal tables. The device server shall remove the corresponding security association from its own tables.

7.7.4.9 Vendor ID payload

An IKE payload type value of 2Bh indicates a Vendor ID payload. This is a protocol extension mechanism. See RFC 4306, except that the paragraph on the topic of Internet-Drafts does not apply to SCSI. The CRIT bit shall be set to zero in a Vendor ID payload.

7.7.4.10 Encrypted payload

An IKE payload type value of 2Eh indicates an Encrypted payload that carries other IKE payloads in Encrypted form. Note that the Next Payload field of the Encrypted payload is the type of the first IKE payload within the Encrypted payload. The Encrypted payload is specified in RFC 4306.

7.7.4.11 SCSI cryptographic algorithms payload

[Editor's note: correct the following number before inclusion in SPC-4]

7.7.4.11.0 SCSI cryptographic algorithms overview

An IKE payload type value of 81h indicates a SCSI Cryptographic Algorithms payload. This payload replaces the IKE Security Association payload.

Table I1 – SCSI Cryptographic Algorithms payload format

Bit Byte	7	6	5	4	3	2	1	0
0	NEXT PAYLOAD							
1	CRIT	RESERVED						
2	(MSB)	PAYLOAD LENGTH (m+1)						
3								(LSB)
SCSI CRYPTOGRAPHIC ALGORITHMS PAYLOAD HEADER								
4	NUMBER OF TRANSFORMS							
5	SECURITY ASSOCIATION USAGE							
6	USAGE DATA LENGTH (k)							
7								

Bit Byte	7	6	5	4	3	2	1	0
8	(MSB)	SAID						
11								(LSB)
12	(MSB)	USAGE DATA						
12+k-1								(LSB)
SCSI CRYPTOGRAPHIC ALGORITHM DESCRIPTORS								
12+k		ALGORITHM DESCRIPTORS (VARIABLE)						
m								

The NEXT PAYLOAD field, CRIT bit, and PAYLOAD LENGTH field are defined in 7.7.4.1.

The NUMBER OF TRANSFORMS field contains the number of algorithm descriptors in the payload.

The SECURITY ASSOCIATION USAGE field shall be set to a value from the following table indicating the purpose of the SA.

The USAGE DATA LENGTH field shall be set to the size of the included usage data. The value of this field shall be a multiple of four, including zero. This value shall conform to the restrictions of the applicable row of the following table.

[Editor's note: Need a table here. Allocate one value for SSC-3 drive keying. Include a column in the table for length of USAGE DATA (range) which should be zero for SSC-3.]

The SAID shall be zero for use of this payload in phase 1. In the Key Exchange phase, the SAID is set to the SAI of the application client.

The USAGE DATA shall contain additional data specified by the command set that specifies how the created security association is to be used.

The algorithm descriptor format is shown in table J1.

Table J1 – SCSI Cryptographic Algorithms Descriptor

Bit Byte	7	6	5	4	3	2	1	0
0	ALGORITHM TYPE							
1	RESERVED							
2	(MSB)	DESCRIPTOR LENGTH (000Ch)						
3								(LSB)
4	(MSB)	ALGORITHM IDENTIFIER						
7								(LSB)
8		ALGORITHM ATTRIBUTES						
11								

ALGORITHM TYPE field identifies the SCSI cryptographic algorithms to which the descriptor applies (see table K1).

Table K1 - ALGORITHM TYPE field values

Code	Description	Reference
00h	Reserved	
01h	Encryption Algorithm (ENCR)	7.7.4.11.1
02h	Pseudo-random Function (PRF)	7.7.4.11.2
03h	Integrity Algorithm (INTEG)	7.7.4.11.3
04h	Diffie-Hellman Group (D-H)	7.7.4.11.4
05h-F0h	Restricted	RFC 4306
F1h-F8h	Reserved	
F9h	IKE Authentication Algorithm (IKE-AUTH)	7.7.4.11.5
FAh-FFh	Reserved	

Algorithm identifier values are defined in the subclauses that describe the algorithm types (see table K1). Vendor-specific algorithm types are prohibited by this standard.

Unless otherwise specified in the subclause that describes an algorithm type, the ALGORITHM ATTRIBUTES field is reserved.

In the IKEv2-SCSI Device Capabilities phase (see 5.13.4.2), this payload is used to report the device server's capabilities. The device server shall include all of the algorithms that it is willing to use with the application client that issued the SECURITY PROTOCOL IN command. If an encryption algorithm is supported with more than one key length, an instance of algorithm data shall be included for each key length. The algorithm data instances shall be ordered by increasing ALGORITHM TYPE, increasing ALGORITHM IDENTIFIER within the same ALGORITHM TYPE, and increasing key length within the same ALGORITHM IDENTIFIER. Failure to observe this ordering may result in Authentication failures because the device server and application client do not agree on the data transferred by the SECURITY PROTOCOL IN command.

In the Key Exchange phase (see 5.13.4.3), this payload is used to inform the device server of the algorithms that the application client has selected. The device server echoes this payload to confirm acceptance of those algorithms. In the Key Exchange phase, this payload shall contain one instance of algorithm data for each of the six values of ALGORITHM TYPE in order of increasing ALGORITHM TYPE. If a combined mode encryption algorithm is selected by the application client, the algorithm data for the integrity ALGORITHM TYPE (i.e., 3) shall contain the NONE integrity algorithm. Otherwise, the N ONE integrity algorithm shall not be used. The IKE Authentication Algorithm descriptor designates the authentication algorithm that the device server shall use. The application client may use any authentication algorithm that the device server accepts (see 7.7.4.11.6).

[Editor's Note: Elliptic curve algorithms are currently only specified for Diffie-Hellman in this proposal. Someone who wants additional algorithms will need to tell us what they want.]

7.7.4.11.1 Encryption Algorithm (ENCR) identifiers

Table K2 shows the algorithm identifier values for the encryption algorithm.

Table K2 – Encryption algorithm identifiers

Value	Description	Support	Reference
0003h	ENCR_3DES	Optional	RFC 2451
000Bh	ENCR_NULL	Mandatory	RFC 2410
000Ch	ENCR_AES_CBC	Mandatory	RFC 3602
000Eh	ENCR_AES_CCM_8	Optional	RFC 4309

Value	Description	Support	Reference
0010h	ENCR_AES_CCM_16	Mandatory	RFC 4309
0014h	AES_GCM with a 16 octet ICV	Optional	RFC 4106
0400h – 0FFFh	Reserved		
1000h – FFFFh	Vendor Specific		
All other values	Restricted		IANA

For encryption algorithm identifiers the ALGORITHM ATTRIBUTES field has the format shown in table ROW2.

Table ROW2 – Encryption algorithm attributes format

Bit Byte	7	6	5	4	3	2	1	0
0	RESERVED							
1	RESERVED							
2	(MSB)	KEY LENGTH						
3								(LSB)

The KEY LENGTH field contains the number of bytes in the key used by the encryption algorithm indicated by the ALGORITHM IDENTIFIER field. For ENCR_3DES, the KEY LENGTH field shall have the value Eh, for AES-based encryption algorithms, the KEY LENGTH field shall have the value 10h or 20h.

NOTE - This forbids 192-bit AES keys; 128-bit and 256-bit keys are allowed.

7.7.4.11.2 Pseudo-random Function (PRF) identifiers

Table K3 shows the algorithm identifier values for the pseudo-random function algorithm.

Table K3 – PRF identifiers

Value	Description	Support	Reference
0002h	PRF_HMAC_SHA1	Mandatory	RFC 2104
0004h	PRF_AES128_CBC	Optional	RFC 4434
0401h ?	PRF_HMAC_SHA256	Optional	RFC 2104?
0402h – 0FFFh	Reserved		
1000 – FFFFh	Vendor Specific		
All other values	Restricted		IANA

[Editor's Note: IETF is in the process of assigning a PRF code for SHA256. That code will need to be supported.]

7.7.4.11.3 Integrity Algorithm (INTEG) identifiers

Table K4 shows the algorithm identifier values for the integrity algorithm.

Table K4 – Integrity algorithm identifiers

Value	Description	Support	Reference
0000h	NONE	Mandatory	RFC 4306
0002h	AUTH_HMAC_SHA1_96	Mandatory	RFC 2404

Value	Description	Support	Reference
0005h	AUTH_AES_XCBC_96	Optional	RFC 3566
0009h	AUTH_AES_128_GMAC	Optional	RFC 4543
000Bh	AUTH_AES_256_GMAC	Optional	RFC 4543
0402h – 0FFFh	Reserved		
1000 – FFFFh	Vendor Specific		
All other values	Restricted		IANA

[Editor's Note: This assumes that SSC-3 use of ESP will keep the 96-bit checksum and the IKEv2 algorithms should be consistent with that. If so, use of HMAC_SHA256 doesn't make much sense here.]

The GMAC integrity algorithms require an Initialization Vector. The Initialization Vector in the Encrypted Payload is used for this purpose, therefore ENCR_NULL shall be used with the GMAC integrity algorithms, and a device server reporting any GMAC integrity algorithm as a device capability shall also report ENCR_NULL as a device capability.

7.7.4.11.4 Diffie-Hellman Group (D-H) identifiers

Table K5 shows the valid Diffie-Hellman algorithm identifiers (i.e., group identifiers) for IKEv2-SCSI. In Table K5, the column entitled "Key Size" indicates the size, in bytes, of the public value within the KEY EXCHANGE DATA field (see 7.7.4.3). A device server should not support finite field Diffie-Hellman groups with less that 2048 bits or elliptic curve fields of less than 256 bits.

Table K5 – Diffie-Hellman group identifiers

Value	Description	Key Size	Support	Reference
0000h – 000Ch	Restricted			IANA
000Dh	2048-bit MODP group (finite field D-H)	256	Mandatory	RFC 3526
000Eh	3072-bit MODP group (finite field D-H)	384	Mandatory	RFC 3526
000Fh – 0012h	Restricted			IANA
0013h	256-bit prime elliptic curve field P-256	32	Optional	NIST FIPS 186-2
0014h	384-bit prime elliptic curve field P-384	48	Optional	NIST FIPS 186-2
0015h	521-bit prime elliptic curve field P-521	66	Optional	NIST FIPS 186-2
0016h – 03FFh	Restricted			IANA
0400h – 0FFFh	Reserved			
1000h - FFFFh	Vendor specific			

[Editor's Note: Is the FIPS 186-2 reference sufficient for all the elliptic curve details?]

7.7.4.11.5 IKE Authentication Algorithm (IKE-AUTH) identifiers

Table K6 shows the algorithm identifier values for the IKE authentication algorithm.

Table K6 – IKE authentication algorithm identifiers

Value	Description	Support	Reference
0000h	IKE_AUTH_NONE	Optional	7.7.4.11.6
0001h	RSA Digital Signature	Optional	RFC 4306
0002h	Shared Key Message Integrity Code	Mandatory	RFC 4306
0003h	DSS Digital Signature	Optional	RFC 4306
0400h – 0FFFh	Reserved		
1000 – FFFFh	Vendor Specific		
All other values	Restricted		IANA

IKE_AUTH_NONE indicates lack of IKEv2-SCSI authentication. If it is reported by a device server in its capabilities and selected by an application client, phase 3 of IKEv2-SCSI is skipped and the resulting SAs are not authenticated. IKE_AUTH_NONE shall not be offered or selected as a default; explicit administrative action shall be required for a device server to offer IKE_AUTH_NONE and for an application client to select it.

Use of certificates with signature-based authentication is optional and determined by presence vs. absence of the optional Certificate and Certificate Request payloads.

The ALGORITHM ATTRIBUTES for IKE Authentication Algorithms are specified in Table L1.

Table L1 – IKE Authentication Algorithms - Attributes

Bit Byte	7	6	5	4	3	2	1	0
0	RESERVED					HTTP	USE	ACCEPT
1	RESERVED							
2	RESERVED							
3	RESERVED							

The HTTP bit indicates whether the device server is capable of using the HTTP protocol to look up certificates. If it is set to zero, the application client shall not use URL-based certificate formats. The HTTP bit shall be set to zero for algorithms that cannot use certificates, including the IKE_AUTH_NULL (see table K6) and IKE_AUTH_SHARED_KEY (see RFC 4306) algorithm identifiers.

> NOTE: The application client uses the CERTIFICATE ENCODING field in the Certificate Request payload to indicate to the device server whether or not a URL-based certificate format is acceptable, see 7.7.4.5.

The USE bit indicates whether the device server is capable of authenticating itself using the authentication algorithm. The USE bit shall be set to one for the IKE_AUTH_NULL algorithm identifier.

The ACCEPT bit indicates whether the device server is capable of validating an application client authentication that uses the authentication algorithm. The ACCEPT bit shall be set to one for the IKE_AUTH_NULL algorithm identifier.

7.7.4.12 SCSI timeout values payload

An IKE payload type value of 82h indicates a SCSI Timeout Values payload. This payload contains timeout values that indicate how long the device server retains state for the IKEv2-SCSI protocol and the SA that it creates

Table M1 – SCSI Timeout Values payload format

Bit Byte	7	6	5	4	3	2	1	0
0	NEXT PAYLOAD							
1	CRIT	RESERVED						
2	(MSB)	PAYLOAD LENGTH (m+1)						
3								(LSB)
4								

Bit Byte	7	6	5	4	3	2	1	0
5	RESERVED							
6								
7	NUMBER OF TIMEOUT VALUES (2)							
8	(MSB)	IKEv2-SCSI PROTOCOL TIMEOUT						
11								(LSB)
12	(MSB)	IKEv2-SCSI SA INACTIVITY TIMEOUT						
15								(LSB)

The NEXT PAYLOAD field, CRIT bit, and PAYLOAD LENGTH field are defined in 7.7.4.1.

The NUMBER OF TIMEOUT VALUES field shall be set to two.

The IKEv2-SCSI PROTOCOL TIMEOUT specifies the number of seconds that the device server shall wait for the next command in the IKEv2-SCSI protocol phase 2 (see 5.13.4.3) or phase 3 (see 5.13.4.4). If the timeout expires before the device server receives the next command, the device server should discard the state for this protocol instance. After the state for a protocol instance is discarded, the device server shall terminate all IKEv2-SCSI protocol commands other than the Security Protocol Out command with SECURITY PROTOCOL SPECIFIC field set to 102h with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to IKEv2-SCSI INVALID COMMAND SEQUENCE.

The IKEv2-SCSI SA INACTIVITY TIMEOUT specifies the number of seconds that the device server shall wait for the next command that uses an SA. This value is copied to the TIMEOUT parameter of the SA created by IKEv2-SCSI.

Zero is a prohibited value for timeouts in an STV payload. If an STV payload is received with any timeout having the value zero, the device server shall terminate the command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST and the additional sense code set to IKEv2-SCSI PARAMETER VALUE INVALID.

7.7.3.5 IKE Errors

Table N1 maps the IKEv2 errors reported via the Notify payload (see Section 3.10.1 of RFC 4306) to additional sense codes.

Table N1 – IKE Errors

IKEv2 Notify Error Type	IKEv2 Description	Sense Key	Additional sense code
0h	Reserved		
1h	UNSUPPORTED_CRITICAL_PAYLOAD	ILLEGAL REQUEST	IKEv2-SCSI PARAMETER NOT SUPPORTED
4h	INVALID_IKE_SPI	ILLEGAL REQUEST	IKEv2-SCSI PARAMETER VALUE INVALID
5h	INVALID_MAJOR_VERSION	ILLEGAL REQUEST	IKEv2-SCSI PARAMETER VALUE INVALID
7h	INVALID_SYNTAX [a]	ILLEGAL REQUEST	IKEv2-SCSI PARAMETER VALUE INVALID
9h	INVALID_MESSAGE_ID	ILLEGAL REQUEST	IKEv2-SCSI PARAMETER VALUE INVALID
Bh	INVALID_SPI	ILLEGAL REQUEST	IKEv2-SCSI PARAMETER VALUE INVALID [b]
Eh	NO_PROPOSAL_CHOSEN [c]	ILLEGAL REQUEST	IKEv2-SCSI PARAMETER VALUE INVALID

IKEv2 Notify Error Type	IKEv2 Description	Sense Key	Additional sense code
11h	INVALID_KE_PAYLOAD [c]	ILLEGAL REQUEST	IKEv2-SCSI PARAMETER VALUE INVALID
18h	AUTHENTICATION_FAILED	ABORTED COMMAND	AUTHENTICATION FAILED
22h - 27h	See RFC 4306 [d]	n/a	n/a
2000h – 3FFFh	Vendor Specific		
All others	Restricted		

[a] RFC 4306 restrictions on when this value is returned for a syntax error within an encrypted payload; do not apply to IKEv2-SCSI.

[b] PARAMETER VALUE INVALID shall be used for an invalid SAID in an IKEv2-SCSI SECURITY PROTOCOL IN or SECURITY PROTOCOL OUT. The additional sense code for an invalid SAID in all other commands is specified by the appropriate command set specification.

[c] The NO_PROPOSAL_CHOSEN and INVALID_KE_PAYLOAD notify error types are replaced by PARAMETER VALUE INVALID because IKEv2-SCSI has a different negotiation structure. As defined in RFC 4306, an IKEv2 initiator shall offer one or more proposals to a responder without knowing what is acceptable to the responder, and shall likewise choose a DH group without knowing whether it is acceptable to the responder; these two notify error types allow the responder to inform the initiator that one or more of its choices are not acceptable. In contrast, an IKEv2-SCSI application client obtains the device server capabilities in the Device Capabilities phase (see 5.13.4.2) and selects algorithms from them in the Key Exchange phase (see 5.13.4.3). An error can only occur if the application client has made an invalid selection, hence the PARAMETER VALUE INVALID description. An application client recovers by restarting processing in the Device Capabilities phase to rediscover the device server's capabilities.

[d] These IKEv2 Error Types correspond to features that are not used in IKEv2-SCSI SA establishment.

[Editor's Note: IKEv2-SCSI PARAMETER VALUE INVALID and IKEv2-SCSI PARAMETER NOT SUPPORTED are new ASC/ASCQ codes; recommend assigning 74h/30h and 74h/31h.]

[Editor's Note: AUTHENTICATION FAILED is a new ASC/ASCQ; recommend assigning 74h/40h.]

If the sense key is ILLEGAL REQUEST, the sense data shall contain a sense key specific sense data descriptor for the ILLEGAL REQUEST sense key that uses the **FIELD POINTER** field to designate the position of the first byte of the first field in the command that caused the error.