To: INCITS T10 Committee

From: Matt Ball, Quantum
David Black, EMC

Date: 23 April 2007

Document: T10/06-449r4

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 ;-).]

Revision 2: Incorporate comments from January Orlando meeting and additional design work. Major changes:

- Use separate SCSI SA Creation Capabilities payload in Device Server Capabilities phase. This removes the erroneous use of Usage Data in the Device Server Capabilities phase.
- Adopt certificate encoding recommendations from RFC 4718 and in addition prohibit Hash and URL certificate formats.
- Add new IANA-allocated values for crypto mechanisms. Remove GMAC as RFC 4543 does not define its use with IKEv2. Adjust vendor-specific ranges to match IANA private use ranges for IKEv2.
- Allow Fibre Channel names as identifiers (for FC-SP certificates).
- Tighten down specification of Delete to reflect removal of Child SAs and avoid problems - it has to be sent on the SA to be deleted because there are no child SAs. Add model clause to explain how Delete works.
- Factor out SA creation command sequencing into a separate subsection. This reduces the amount of text and covers a number of additional error cases.
- Add subsection on how to populate the fields of an SA.
- Renumber IKEv2-SCSI exchange types into IKEv2's private use range. Add additional explanation of IKEv2 header fields, including sequence association checks and errors.
- Lots of other edits and changes (e.g., to remove Editor's notes).

Revision 3: Incorporate comments from March Memphis meeting. Major changes:

- Remove DSS digital signature support. Remove support for encryption without integrity. Finish aligning cryptographic algorithm identifiers with IANA registries for IKEv2.
- Terminology change to SA creation cryptographic command sequence, only allow one at a time per I_T_L Nexus (so device server only has to save one set of active parameters), but return NOT READY if another is attempted instead of aborting the original sequence.
- Adapt to USAGE changes made to SA proposal as part of approving it.
- Add key length values to encryption algorithm table. Add notes about extra salt bytes that CCM and GCM mode take from KEYMAT.

Revision 4: Incorporate comments from April Houston interim meeting. Major changes:

- Change terminology from protocol "phases" to protocol "steps", add summary of protocol steps.
- Distinguish IKEv2-SCSI keys from SA keys for clarity.
- Modify (generally reduce) allowed cryptographic algorithms. GMAC cannot be added because IETF does not support GMAC usage in IKEv2.
- Add AUTH_NONE integrity support (no separate integrity algorithm) for use with AES_CCM and AES_GCM combined mode algorithms that provide integrity.
- Expand SA type (usage) to 2 bytes and reformat SCA payload accordingly. Did not add a second pair of nonces because IKEv2 (RFC 4306) uses the same nonces for the IKEv2 (SA) keys and the keys for the first child SA.

2 References

T10/SSC-3r3c, SCSI-3 Stream Commands.
T10/SPC-4r10, SCSI Primary Commands.
T10/06-225r5 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.
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.

3.1 Differences between IKEv2 and IKEv2-SCSI

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

a) An SA created by the IKEv2-SCSI protocol is used to directly protect SCSI traffic. There is no concept of child SAs; this is based on a design assumption that SA usage will be infrequent in SCSI command streams.
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, in part because 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 use 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 180-2 *Secure Hash Standard*
NIST FIPS 198a, *The Keyed-Hash Message Authentication Code (HMAC)*

2.5 IETF References

RFC 2404, *The Use of HMAC-SHA-1-96 within ESP and AH*
RFC 2410, *The NULL Encryption Algorithm and Its Use With IPsec.*
RFC 2437, *PKCS #1: RSA Cryptography Specifications Version 2.0*
RFC 3280, *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*
RFC 3447, *Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1*
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.*
RFC 4753, *ECP Groups for IKE and IKEv2*
RFC 4754, *IKE and IKEv2 Authentication Using the Elliptic Curve Digital Signature Algorithm (ECDSA)*

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 keys: Keys used to provide security for IKEv2-SCSI operations.
3.1.c IKEv2-SCSI SA creation cryptographic command sequence (CCS): A sequence of Key Exchange step and Authentication step (if used) SECURITY PROTOCOL IN and SECURITY PROTOCOL OUT commands that are part of an IKEv2-SCSI SA creation transaction (see 5.13.4).
3.1.d SA creation transaction: A sequence of SECURITY PROTOCOL IN commands and SECURITY PROTOCOL OUT commands used to create an SA between an application client and device server (see 5.13.4).
3.1.e SA generation: Computation and initialization of the SA parameter values required to create a security association. This is the final step in creating a security association after all SCSI commands required to create a security association have been performed.
3.1.f SA keys: Keys used by an SA to provide security for the operations that use the SA.
3.1.g SA Participant: An application client or device server that participates in the creation or use of an SA.

3.2 Symbols and acronyms

IKEv2-SCSI CCS	IKEv2-SCSI Cryptographic Command Sequence (see 3.1.b)
PKI	Public Key Infrastructure (see RFC 3280)

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 creation capabilities	6.27.3
xxh	IKEv2-SCSI	5.13.4

...

5.13.4 Using IKEv2-SCSI to create a security association

5.13.4.1 Using IKEv2-SCSI to create a security association overview

The IKEv2-SCSI protocol is a subset of the IKEv2 protocol (see RFC 4306) that this standard defines for use in the creation and maintenance of an SA (see 5.13.1).

An IKEv2-SCSI SA creation transaction occurs between an application client and a device server, and shall only be initiated by the application client.

The IKEv2-SCSI protocol creates two SAs:

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

IKEv2-SCSI SA creation encompasses up to three steps that shall be performed in the following order:

1. **Device Server Capabilities step** (see 5.13.4.2): The application client determines the device server's cryptographic capabilities;
2. **Key Exchange step** (see 5.13.4.3): The application client and device server:
   a. Perform a key exchange;
   b. Determine SAIs; and
   c. May complete the creation of the SA; and
3. **Authentication step** (see 5.13.4.4): Unless omitted by application client and device server negotiations in the previous steps:
   a. The application client and device server authenticate:
      a. Each other;
      b. The key exchange; and
      c. The capability selection; and
   b. Complete the creation of the SA.

The values in the SECURITY PROTOCOL field and the SECURITY PROTOCOL SPECIFIC field in the SECURITY PROTOCOL IN command and SECURITY PROTOCOL OUT command identify the step for the IKEv2-SCSI protocol (see 7.7.4.1).

The Key Exchange step and the Authentication step depend on the results from the Device Capabilities step in order to create an SA. Two sets of keys are involved in creation of an SA:

1. IKEv2-SCSI keys that are created by the IKEv2-SCSI Key Exchange step. These keys are used by the IKEv2-SCSI Authentication step and to delete the SA.
2. SA keys created as part of generating the SA. These keys are used by SCSI operations that obtain security from the generated SA.

An application client may or may not:

a) Proceed to the Key Exchange step after the Device Server Capabilities step; or
b) Perform a separate Device Server Capabilities step for each IKEv2-SCSI SA creation transaction.

If the device server's capabilities have changed, the Key Exchange step may return an error, and the Authentication step shall return an error. The application client may recover from such errors by repeating the Device Server Capabilities step.

After a Device Capabilities step, the application client performs SA creation by issuing a sequence of four IKEv2-SCSI commands. These four commands constitute an IKEv2-SCSI CCS (see 3.1.c):

1) A Key Exchange step SECURITY PROTOCOL OUT command
2) A Key Exchange step SECURITY PROTOCOL IN command
3) An Authentication step SECURITY PROTOCOL OUT command
4) An Authentication step SECURITY PROTOCOL IN command

If the application client and device server agree to use IKE_AUTH_NONE, the Authentication step is skipped and in this case the IKEv2-SCSI CCS consists of the two Key Exchange step commands.

Use of the Authentication step is negotiated in the Device Server Capabilities step and the Key Exchange step. If both SA participants agree that the Authentication step is not used, then the Authentication step is omitted and SA creation occurs upon the completion of the Key Exchange step. If either SA participant requires that the Authentication step be used, the application client shall initiate the Authentication step following the Key Exchange step.

SA participants should perform the Authentication step unless man-in-the-middle attacks (see 5.13.1.4) are not of concern or are prevented by other means such as physical security of the transport.

> NOTE – The Authentication step should be performed if there is any doubt as to whether it is needed. Omission of the Authentication step provides no defense against a man-in-the-middle adversary that is capable of modifying SCSI commands, as such an adversary can insert itself as an intermediary on the created SA without knowledge of the SA participants, thereby completely subverting the intended security. Omission of the Authentication step is only appropriate in environments where the absence of such adversaries is assured by other means, for example, a direct physical connection between the systems on which the application client and device server or use of end-to-end security in the SCSI transport security such as IPsec for iSCSI.

The Key Exchange step commands and Authentication step commands (if the Authentication step is not omitted) that create a single SA shall be performed in order on the same I_T_L Nexus.

5.13.4.1.1 IKEv2-SCSI Protocol Summary

[Editor's Note: This section will get reformatted to the "lozenges with arrows" diagram format discussed in Houston. In all cases, the Application Client is on the left and the Device Server is on the right.]

The IKEv2-SCSI protocol employs a header and payloads for parameter data in IKEv2-SCSI SECURITY PROTOCOL OUT and SECURITY PROTOCOL IN commands. The following acronyms and notation are used to represent the header and payloads in this section:

• AUTH	Authentication payload
• CERT	Certificate payload
• CERTREQ	Certificate Request payload
• E { A, B, ... }	Encrypted payload containing the A, B, ... payloads.
• HDR	IKEv2-SCSI Header
• ID	Identification

KE	Key Exchange
N-IC	Notify payload carrying an Initial Contact notification
NONCE	Nonce
SCA	SCSI Cryptographic Algorithms
SSCC	SCSI SA Creation Capabilities
STV	SCSI Timeout Values

[Editor's Note: This should probably be a table, and should cross reference Table G1 in 7.7.4.1]

Square brackets (i.e., [ ]) are used to enclose optional payloads.

The Device Server Capabilities step consists of a SECURITY PROTOCOL IN command carrying a SCSI SA Creation Capabilities payload. The IKEv2-SCSI header (HDR) is not used:

Application Client Device Server
← SSCC —

The SSCC payload indicates the device server's capabilities for SA creation.

The Key Exchange step consists of a SECURITY PROTOCOL OUT command followed by a SECURITY PROTOCOL IN command conveying the parameter data indicated in the diagram:

Application Client Device Server
— HDR, STV, SCA, KE, NONCE →
← HDR, SCA, KE, NONCE, [CERTREQ] —

The STV payload contains timeouts for SA creation and usage. The SCA payloads select and agree on usage of the SA and the cryptographic algorithms for the SA. The KE and NONCE payloads are part of the key and nonce exchanges that are used to generate SA keys. The optional CERTREQ payload enables the device server to request a certificate from the application client.

The Authentication step is realized by a SECURITY PROTOCOL OUT command followed by a SECURITY PROTOCOL IN command conveying the parameter data indicated in the diagram:

Application Client Device Server
— HDR, E { ID, [CERT], [CERTREQ], [N-IC], AUTH } →
← HDR, E { ID, [CERT], AUTH } —

All payloads in the Authentication step are protected using the cryptographic algorithms determined by the SCA payloads in the Key Exchange step.

The ID payloads contain the identities to be authenticated; these identities are not required to be SCSI names or identities. The optional CERT payloads contain certificates for authentication, and the optional CERTREQ payload enables the application client to request a certificate from the device server. The optional N-IC payload provides a mean to tear down stale SAs between the same SA participants. The AUTH payloads authenticate not only the SA participants, but also the entire protocol sequence (e.g., the AUTH payloads prevents a man-in-the-middle attack from succeeding).

5.13.4.2 Device Server Capabilities step

In the Device Server Capabilities step, the application client sends a SECURITY PROTOCOL IN command with the SECURITY PROTOCOL field set to SA Creation Capabilities (i.e., zzh) and the SECURITY PROTOCOL SPECIFIC field set to 0101h (i.e., read the device server's IKEv2-SCSI cryptographic capabilities) (see 6.27.3).

The device server shall use the authentication algorithm type F9h (see 7.7.4.6) in the Device Server Capabilities step to report supported IKEv2-SCSI authentication algorithms.

An authentication algorithm type of IKE_AUTH_NONE is used to indicate that the device server permits the Authentication step to be omitted (see 5.13.4.1). The device server shall not return IKE_AUTH_NONE value as an authentication algorithm type in the Device Server Capabilities step unless the device server has been explicitly configured to do so by means outside the scope of this standard.

> NOTE – The Device Server Capabilities step has no IKEv2 exchange equivalent in RFC 4306. This step replaces most of IKEv2's negotiation by having the application client obtain the supported capabilities from the device server.

5.13.4.3 Key Exchange step

5.13.4.3.1 Key Exchange step overview

The Key Exchange step consists of an unauthenticated Diffie-Hellman key exchange with nonces (see RFC 4306) and is accomplished as follows:

1) A SECURITY PROTOCOL OUT command (see 5.13.4.3.2); and
2) A SECURITY PROTOCOL IN command (see 5.13.4.3.3).

> NOTE – The Key Exchange step corresponds to the IKEv2 IKE_SA_INIT exchange in RFC 4306, except that determination of device server capabilities has been moved to the Device Server Capabilities step.

5.13.4.3.2 Key exchange step SECURITY PROTOCOL OUT command

To send its key exchange parameters to the device server, the application client sends a SECURITY PROTOCOL OUT command with the SECURITY PROTOCOL field set to IKEv2-SCSI (i.e., xxh) and the SECURITY PROTOCOL SPECIFIC field set to 0102h. The parameter data consists of the IKEv2-SCSI header (see 7.7.3.1) and the following:

1) A SCSI Timeout Values (i.e., STV) payload (see 7.7.4.13);
2) A SCSI Cryptographic Algorithms (i.e., SCA) payload (see 7.7.4.12);
3) A Key Exchange (i.e., KE) payload (see 7.7.4.3); and
4) A Nonce (i.e.; NONCE) payload (see 7.7.4.7).

The STV payload contains the inactivity timeouts that apply to this IKEv2-SCSI SA creation transaction and the SA that is created.

The SCA payload contains the cryptographic algorithms selected by the application client and the usage of the created SAs. The cryptographic algorithms shall be selected from the algorithms obtained from the device server in the Device Server Capabilities step (see 5.13.4.2). If the application client is unable to select a set of algorithms that are appropriate for the intended usage of the SA, the application client shall not perform the Key Exchange step and shall not create an SA.

The KE payload contains the application client's Diffie-Hellman value.

The NONCE payload contains the application client's random nonce.

5.13.4.3.3 Key Exchange step SECURITY PROTOCOL IN command

If the Key Exchange step SECURITY PROTOCOL OUT command completes with GOOD status, then the application client shall send a SECURITY PROTOCOL IN command with the SECURITY PROTOCOL field set to IKEv2-SCSI (i.e., xxh) and with the SECURITY PROTOCOL SPECIFIC field set to 0102h to obtain the device server's key exchange message. The parameter data returned by the device server shall contain the IKEv2-SCSI header (see 7.7.3.1), and the following:

1) A SCSI Cryptographic Algorithms (i.e., SCA) payload (see 7.7.4.12);
2) A Key Exchange (i.e., KE) payload (see 7.7.4.3);
3) A Nonce (i.e., NONCE) payload (see 7.7.4.7); and
4) Zero or more Certificate Request (i.e., CERTREQ) payloads (see 7.7.4.5).

During the processing of the Key Exchange step SECURITY PROTOCOL IN command, the device server shall:

a) Associate the SECURITY PROTOCOL IN command to the most recently processed Key Exchange step SECURITY PROTOCOL OUT command received on the I_T_L nexus;
b) Return the cryptographic algorithms supplied by the application client in the Key Exchange step SECURITY PROTOCOL OUT command;
c) Return its SAI in the SCA Payload;
d) Return information about the completed the Diffie-Hellman exchange with the KE Payload; and
e) Return its nonce in the NONCE Payload.

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

The application client shall compare the cryptographic algorithms and usage data received in the SCA payload of the Key Exchange step SECURITY PROTOCOL IN command to the cryptographic algorithms and usage data that the application client sent in the SCA payload of the Key Exchange step SECURITY PROTOCOL OUT command. If the algorithms and usage data are not the same, the application client shall report an error in a vendor-specific manner, shall not complete the Key Exchange step (see 5.13.4.3.4), shall not perform the Authentication step (see 5.13.4.4) and shall not create the SA (see 5.13.4.5).

5.13.4.3.4 Key Exchange step completion

After the Key Exchange step SECURITY PROTOCOL IN command completes with GOOD status, the SA participants shall:

a) generate SKEYSEED (see RFC 4306) using the specified pseudo-random function before proceeding to the Authentication step, if applicable; and then
b) use SKEYSEED to generate the following IKEv2-SCSI keys (see RFC 4306):

    A) SK_d: A key used to generate the SA keys. 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 in generation of keyed MACs at the application client (SK_ai) and the device server (SK_ar). 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 encryption at the application client (SK_ei) and the device server (SK_er) 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 (i.e., F9h) for the authentication algorithm type in the Key Exchange step, and the Key Exchange step completes without errors, the application client shall not perform the Authentication step. In this case, both the application client and the device server shall generate the SA as specified in 5.13.4.5.

The application client shall not select the IKE_AUTH_NONE value as an authentication algorithm type in the Key Exchange step unless:

a. It is present in the parameter data returned during the Device Server Capabilities step; and
b. The application client is configured to omit the Authentication step by an administrator.

NOTE – When IKE_AUTH_NONE is used, IKEv2-SCSI has no protection against any man-in-the-middle attacks. The following administrative decisions are security policy decisions that absence of authentication is acceptable, and should only be made with a full understanding of the security consequences of the lack of authentication:
a) enabling return of the IKE_AUTH_NONE authentication algorithm type in the Device Capabilities step, and
b) enabling the application client to select IKE_AUTH_NONE in the Key Exchange step
Such decisions should only be made in situations where active attacks on IKEv2-SCSI are not of concern (e.g., direct attachment of initiator and target, end-to-end secure transport channel such as IPsec for iSCSI) .

5.13.4.4 Authentication step

5.13.4.4.1 Authentication step overview

The Authentication step performs the following functions:
a) authenticates both the application client and the device server;
b) protects the previous steps of the protocol; and
c) cryptographically binds the authentication and the previous steps to the generated SA.

The Authentication step is accomplished as follows:
1. A SECURITY PROTOCOL OUT command with the SECURITY PROTOCOL field set to IKEv2-SCSI (i.e., xxh) and the SECURITY PROTOCOL SPECIFIC field set to 0103h (see 5.13.4.4.2); and
2. A SECURITY PROTOCOL IN command with the SECURITY PROTOCOL field set to IKEv2-SCSI and the SECURITY PROTOCOL SPECIFIC field set to 0103h (see 5.13.4.4.3).

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

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

5.13.4.4.2 Authentication step SECURITY PROTOCOL OUT command

The application client sends a SECURITY PROTOCOL OUT command with the SECURITY PROTOCOL field set to IKEv2-SCSI (i.e., xxh) and the SECURITY PROTOCOL SPECIFIC field set to 0103h to send its authentication information to the device server. This parameter data consists of the IKEv2-SCSI header (see 7.7.3.1) and an encrypted payload (i.e., E see 7.7.4.10) that contains the following:
1. An Identification – Application Client (i.e., ID) payload (see 7.7.4.4);
2. Zero or more Certificate (i.e., CERT) payloads (see 7.7.4.5);
3. Zero or more Certificate Request (i.e., CERTREQ) payloads (see 7.7.4.5);
4. Zero or one Notify (i.e., N-IC), payload (see 7.7.4.8a); and
5. An Authentication (i.e., AUTH) payload (see 7.7.4.6).

The application client shall:
a) assert its identity with the ID payload;
b) prove knowledge of the secret corresponding to ID; and
c) integrity protect the prior steps using the AUTH Authentication payload.

The application client may send its Certificate(s) in CERT payload(s) as described in RFC 4306. The application client may send a list of its trust anchors in CERTREQ payload(s) as described in RFC 4306. If any CERT payloads are included in the parameter data, the first CERT payload shall contain the public key used to verify the Authentication (i.e., AUTH) payload.

The application client and device server may use different authentication methods, so the use of CERT and CERTRQ payloads may differ between the Authentication step SECURITY PROTOCOL OUT command and the Authentication step SECURITY PROTOCOL IN command.

The application client uses the Notify (i.e., N) payload to send an Initial Contact notification to the device server. The Initial Contact notification informs the device server that this newly created IKEv2-SCSI SA should 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 ID payload contents that are identical to those in previous Key Exchange step SECURITY PROTOCOL OUT commands. The device server shall delete other SAs only after the completion of both steps in the Authentication step (i.e., the device server shall ignore a Notify payload that includes an Initial Contact notification if an error occurs during the Authentication step).

The device server shall verify the AUTH payload as defined in 7.7.4.6. The CERT payload(s) are used as part of this verification for PKI-based authentication. If the device server is unable to proceed with SA creation for any reason (e.g., the verification of the AUTH payload fails), 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 an appropriate value. The additional sense code AUTHENTICATION FAILED shall be used when verification of the AUTH payload fails, or when authentication fails for any other reason.

[Editor's Note: AUTHENTICATION FAILED is a new ASC/ASCQ]

5.13.4.4.3 Authentication step SECURITY PROTOCOL IN command

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 set to 0103h to obtain the device server's authentication information. The parameter data consists of the IKEv2-SCSI header (see 7.7.3.1), and an encrypted payload, (i.e., E, see 7.7.4.10), that contains the following:

1. An Identification – Device Server (i.e., ID) payload (see 7.7.4.4);
2. Zero or more Certificate (i.e., CERT) payloads (see 7.7.4.5); and
3. A receiver Authentication (i.e., AUTH) payload (see 7.7.4.6).

The device server:

a) asserts its identity with the ID payload;
b) authenticates its identity; and
c) protects the integrity of the prior step messages with the AUTH payload.

The device server may return its Certificate(s) in CERT payload(s) as described in RFC 4306. If any CERT payloads are included in the parameter data, the first CERT payload shall contain the public key used to verify the Authentication (i.e., AUTH) payload.

The application client and device server may use different authentication methods, so the use of CERT and CERTRQ payloads may differ between the Authentication step SECURITY PROTOCOL OUT command and the Authentication step SECURITY PROTOCOL IN command.

The application client shall verify the AUTH payload as defined in 7.7.4.6. The CERT payload(s) are used as part of this verification for PKI-based authentication. If the AUTH payload is verified and no other error occurs the application client shall create the SA as specified in 5.13.4.5.

If the application client is unable to proceed with SA creation for any reason (e.g., the verification of the AUTH payload fails), the application client:

a) Shall not use the SA for any additional activities; and
b) Shall delete the SA pair as specified in 5.13.5.

5.13.4.5 Security Association generation

The application client and the device server shall initialize the security association (see 5.13.2) as follows (see 5.13.2.2 and 5.13.2.3):

a) AC_SAI shall be set to the value of the IKE_SA APPLICATION CLIENT SAI in the IKEv2-SCSI header of the Key Exchange step SECURITY PROTOCOL OUT command (see 7.7.3.1.1). DC_SAI shall be set to the value of the IKE_SA DEVICE SERVER SAI in the IKEv2-SCSI header of the Key Exchange step SECURITY PROTOCOL IN command (see 7.7.3.1.1).
b) TIMEOUT shall be set to the IKEV2-SCSI SA INACTIVITY TIMEOUT from the STV payload in the Key Exchange step SECURITY PROTOCOL OUT command (see 7.7.4.13).
c) AC_NONCE shall be set to the value of the NONCE DATA field in the NONCE payload sent by the application client in the Key Exchange step SECURITY PROTOCOL OUT command (see 7.7.4.7 and RFC 4306). DS_NONCE shall be set to the value of the NONCE DATA field in the NONCE payload received from the device server in the Key Exchange step SECURITY PROTOCOL IN command (see 7.7.4.7 and RFC 4306).
d) KDF_ID shall be set to the value obtained by changing the two most significant bytes from 0000h to 0002h in the 4-byte ALGORITHM IDENTIFIER field in the SCSI Cryptographic Algorithm descriptor for the PRF algorithm type in the SCA payload sent by the application client in the Key Exchange step SECURITY PROTOCOL OUT command (see 7.7.4.11.1).
e) USAGE_TYPE shall be set to the value of the SECURITY ASSOCIATION TYPE field in the SCA payload sent by the application client in the Key Exchange step SECURITY PROTOCOL OUT command (see 7.7.4.11.1).
f) USAGE_DATA shall contain the collection of the following values from of the USAGE DATA field in the SCA payload sent by the application client in the Key Exchange step SECURITY PROTOCOL OUT command (see 7.7.4.11.1):
    A) the USAGE DATA field,
    B) the ALGORITHM IDENTIFIER and KEY LENGTH fields in the SCSI Cryptographic Algorithm descriptor for the ENCR algorithm type and
    C) the ALGORITHM IDENTIFIER field in the SCSI Cryptographic Algorithm descriptor for the INTEG algorithm type.

    NOTE – The inclusion of the algorithm identifiers and key length in USAGE_DATA for the SA enables the SA to use the same encryption and integrity algorithms as IKEv2-SCSI negotiated for its own use.

g) MGMT_DATA shall be set to the following collection of data necessary for the application client to create IKEv2-SCSI command to delete the SA, and for the device server to check the correctness of such a command (see 5.13.5):
    A) The ALGORITHM IDENTIFIER and KEY LENGTH fields in the SCSI Cryptographic Algorithm descriptor for the ENCR algorithm type in the SCA payload sent by the application client in the Key Exchange step SECURITY PROTOCOL OUT command (see 7.7.4.11.1)
    B) The SK_ei key used for encryption by the application client and for decryption by the device server.
    C) The ALGORITHM IDENTIFIER field in the SCSI Cryptographic Algorithm descriptor for the INTEG algorithm type in the SCA payload sent by the application client in the Key Exchange step SECURITY PROTOCOL OUT command (see 7.7.4.11.1).
    D) The SK_ai key used to compute the cryptographic integrity check at the application client and to verify this check at the device server.
    E) The next value of the MESSAGE ID field in the IKEv2-SCSI header.
h) KEY_SEED shall be set to the value of SK_d computed as part of Key Exchange step completion (see 5.13.4.3.4).

5.13.4.6 IKEv2-SCSI CCS

A new CCS shall be initiated by a Key Exchange step SECURITY PROTOCOL OUT command that is not terminated with CHECK CONDITION and the device server shall allocate a new device server SAI as part of processing that command. The application client shall issue the commands in an IKEv2-SCSI CCS in order on the same I_T_L Nexus. The IKEv2-SCSI PROTOCOL TIMEOUT in the STV payload of the Key Exchange step SECURITY PROTOCOL OUT command shall be the amount of time that the device server is required to wait for the next command in the IKEv2-SCSI CCS; the application client should ensure that this timeout does not expire before it issues that next command. The device server maintains information for SA creation (see 5.13.4.5) while an IKEv2-SCSI CCS is in progress.

The CCS is identified by the device server SAI and the application client SAI received in the Key Exchange step SECURITY PROTOCOL out command. The CCS for the Authentication step SECURITY PROTOCOL OUT command is determined by matching the SAIs in the IKEv2-SCSI header with these two SAIs. The two SECURITY PROTOCOL IN commands are associated with the CCS identified by the preceding SECURITY PROTOCOL OUT command on the same I_T_L nexus and shall return device server parameter data for that CCS; the application client shall check that the received parameter data is for the correct CCS. If any command that is part of a CCS is terminated with CHECK CONDITION status for any reason, that command shall not advance the CCS to expect the next command in the sequence.

The device server shall check the IKEv2-SCSI header and the INTEGRITY CHECKSUM DATA in the Encrypted payload of an Authentication phase SECURITY PROTOCOL OUT command before performing any checks of data contained in the Encrypted payload. If the contents of the IKEv2-SCSI header or the INTEGRITY CHECKSUM DATA in the Encrypted payload cause the Authentication phase SECURITY PROTOCOL OUT command to be terminated with CHECK CONDITION status, the terminated command shall have no effect on the CCS in progress. If the contents of the Encrypted payload cause the Authentication phase SECURITY PROTOCOL OUT command to be terminated with CHECK CONDITION status with a sense key other than NOT READY, the CCS shall be terminated and the SA shall not be generated.

An IKEv2-SCSI CCS shall be terminated and the SA shall not be generated when the IKEv2-SCSI PROTOCOL TIMEOUT expires before the next command that is part of the CCS is received, and the device server discards the SA creation information in response to the timeout expiration. If multiple SECURITY PROTOCOL IN commands for the same phase are received, this timeout shall be measured from completion of the first instance of the command.

> NOTE – Termination of a CCS in response to terminating a SECURITY PROTOCOL IN command with CHECK CONDITION status would create a security vulnerability due to the absence of security checks on CDB for the SECURITY PROTOCOL IN command. If the application client is unable to issue a SECURITY PROTOCOL IN command that is not terminated with CHECK CONDITION status, the CCS will time out, and this is the only means of terminating the CCS.

If any command that is part of an IKEv2-SCSI CCS is received on an I_T_L Nexus while a different IKEv2-SCSI CCS is in progress, that command shall be terminated with CHECK CONDITION status, with the sense key set to NOT READY, and the additional sense code set to IKEv2-SCSI 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 IKEv2-SCSI CCS (e.g., the IKEv2-SCSI CCS is 25% complete if one command out of four has been performed). 5.13.7 applies to any PROGRESS INDICATION that reports progress information beyond the number of commands processed.

[Editor's Note: IKEv2-SCSI OPERATION IN PROGRESS is a new ASC/ASCQ - suggest 00h/1Eh]

If an application client abandons an incomplete IKEv2-SCSI CCS, the protocol timeout in case a) above enables the device server to discard its SA creation information. The device server shall enforce a maximum value for the protocol timeout.

> NOTE: The maximum value for the protocol timeout should be long enough to allow the application client to continue the IKEv2-SCSI CCS, but short enough that if an incomplete IKEv2-SCSI CCS is abandoned, the device server will discard the state for that IKEv2-SCSI CCS and become available to for another IKEv2-SCSI CCS without excessive delay.

If an application client abandons a complete IKEv2-SCSI CCS (e.g., due to an Authentication failure at step 4) above or a parameter data error at step 2) above when the Authentication step is omitted, the device server will create an SA that will never be used. This orphan SA can be removed via use of the SA inactivity timeout in the STV payload to detect that the SA is not being used (see 7.7.4.13). This is a consideration in selection of SA inactivity timeout values.

5.13.5 Deleting an IKEv2-SCSI SA

When an SA is deleted, both sets of SA parameters (see 5.13.2.1) shall be deleted as follows:

1. The application client shall generate an IKEv2-SCSI Delete command that indicates deletion of the application client's SA
2. The application client shall delete its SA.
3. The application client shall send the IKEv2-SCSI Delete command generated in step 1.
4. The device server shall delete the corresponding SA when it processes the Delete command.

The IKEv2-SCSI Delete command shall be a SECURITY PROTOCOL OUT command with the SECURITY PROTOCOL field set to IKEv2-SCSI (i.e., xxh) and the SECURITY PROTOCOL SPECIFIC field set to 0104h. The parameter data for this command shall consist of the IKEv2-SCSI header (see 7.7.3.1) and an encrypted payload (i.e., E see 7.7.4.10) that contains one Delete payload. The Delete payload shall specify the SAI of the SA that the application client has deleted. The Encrypted Payload shall be constructed from on the MGMT_DATA in the device server SA that corresponds to the SA that the application client has deleted. The device server shall delete this SA when it processes a correct Delete command. The application client shall not issue a SECURITY PROTOCOL IN command to obtain a device server response to an IKEv2-SCSI Delete command.

5.13.6 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) Any SECURITY PROTOCOL IN command with an allocation length of up to 16 384 bytes are not terminated with an error due to the number of bytes to be transferred;
b) Any SECURITY PROTOCOL OUT command with a transfer length of up to 16 384 bytes are not terminated with an error due to the number of bytes transferred;
c) The timeout and retransmission mechanisms defined in RFC 4306 shall not be used by application clients and device servers (i.e., retransmission is performed by the applicable SCSI transport protocol);
d) Each SCSI command used by IKEv2-SCSI completes by conveying a status from the device server to the application client;
e) IKEv2-SCSI uses the Message ID field for sequencing (see RFC 4306), but only for SA creation (see 7.7.3.3);
f) IKEv2-SCSI requests shall not be overlapped. If an application client attempts to overlap IKEv2-SCSI requests (see RFC 4306), 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 IKEv2-SCSI PARAMETER VALUE INVALID (see 5.13.4 and 7.7.3.5);
g) 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;
h) IKEv2 cookies (see RFC 4306) are not used in IKEv2-SCSI;
i) IKEv2 cryptographic algorithm negotiation (see RFC 4306) is replaced by the IKEv2-SCSI framework (see 5.13.4, 7.7.4.11, and 7.7.4.12) (i.e., the IKEv2 proposal construct is not used in IKEv2-SCSI);

j) An IKEv2-SCSI SA is rekeyed by replacing it with a new SA:
   a) CHILD_SAs are not used in IKEv2-SCSI;
   b) The RFC 4306 discussion of CHILD_SAs does not apply to IKEv2-SCSI;
   c) Coexistence of the original SA and the new SA that is created for rekeying purposes should be supported; and
   d) IKEv2 does not support rekeying notification for IKE_SAs, therefore IKEv2-SCSI does not support rekeying notification;
k) Traffic Selectors (see RFC 4306) are not used by IKEv2-SCSI;
l) The requirements in RFC 4306 on nonces are be followed for nonces used in IKEv2-SCSI;
m) The RFC 4306 requirements on address and port agility are specific to the user datagram protocol and the IP protocol and does not apply to IKEv2-SCSI;
n) Diffie-Hellman exponential reuse and reuse of analogous Diffie-Hellman public values for Diffie-Hellman mechanisms not based on exponentiation are 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 (see RFC 999999);
o) Keys for the Authentication step of IKEv2-SCSI are generated as specified in RFC 4306;
p) IKEv2-SCSI uses a slightly modified version of the authentication calculations in RFC 4306 (see 7.7.4.6);
q) The RFC 4306 sections that describe the following features are not used in IKEv2-SCSI:
   a) Extensible authentication protocol methods;
   b) Generating keying Material for CHILD_SAs;
   c) Rekeying an IKE SA using CREATE_CHILD_SA;
   d) Requesting an internal address;
   e) Requesting the peer's version;
   f) IPComp;
   g) NAT traversal; and
   h) Explicit congestion notification; and
r) IKEv2 Error Handling (see RFC 4306) is replaced by the use of CHECK CONDITION status and sense data in IKEv2-SCSI (see 5.13.4 and 7.7.3.5).

5.13.7 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, then 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 cryptographic computations that may take a variable amount of time based on their inputs. The device server may use the PROGRESS INDICATION to report synthetic progress that does not reveal the actual progress of the computation (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).

The restrictions in this subclause apply 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, and reveal information about the inputs.

When cryptographic calculations are in progress, the sense data specified in this subclause shall be returned in response to a REQUEST SENSE command.

6.29 SECURITY PROTOCOL IN command

Editor's Note: Modify Table 186 "SECURITY PROTOCOL field in SECURITY PROTOCOL IN command" as follows:

Code	Description	Reference
00h	Security protocol information	~~6.29.2~~ 7.7.1
01h - 06h	Defined by the TCG	3.1.132
07h - 1Fh	Reserved	
20h	Tape Data Encryption	SSC-3
21h – ~~ED~~3Fh	Reserved	
zzh (i.e., 40h)	SA Creation Capabilities	7.7.2
xxh (i.e., 41h)	IKEv2-SCSI	7.7.3
42h - EDh	Reserved	
EEh	Authentication in Host Attachments of Transient Storage Devices	IEEE 1667
EFh	ATA Device Server Password Security	TBD
F0h - FFh	Vendor Specific	

6.30 SECURITY PROTOCOL OUT command

Editor's Note: Modify Table 191 "SECURITY PROTOCOL field in SECURITY PROTOCOL OUT command" as follows:

Code	Description	Reference
00h	Reserved	
01h - 06h	Defined by the TCG	3.1.132
07h - 1Fh	Reserved	
20h	Tape Data Encryption	SSC-3
21h – ~~ED~~40h	Reserved	
xxh (i.e., 41h)	IKEv2-SCSI	7.7.5
42h - EDh	Reserved	
EEh	Authentication in Host Attachments of Transient Storage Devices	IEEE 1667
EFh	ATA Device Server Password Security	TBD
F0h - FFh	Vendor Specific	

...

7.7 Security protocol parameters

7.7.1 Security protocol information parameters

[Editors Note: Move the contents of SPC-4 r07 6.29.2 to here]

7.7.2 SA Creation Capabilities description

7.7.2.1 Overview

The purpose of the SA creation capabilities security protocol (i.e., the SECURITY PROTOCOL field set to zzh in a SECURITY PROTOCOL IN command) is to transfer SA creation related information from the device server. A SECURITY PROTOCOL IN command in which the SECURITY PROTOCOL field is set to zzh is not associated with an previous SECURITY PROTOCOL OUT command and shall be processed without regard for whether a SECURITY PROTOCOL OUT command has been processed.

If SA creation (see 5.13.2.3) is supported, the SECURITY PROTOCOL value of zzh shall be supported as defined in this standard.

7.7.2.1a CDB description

When the SECURITY PROTOCOL field is set to SA Creation Capabilities (i.e., zzh) in a SECURITY PROTOCOL IN command, the SECURITY PROTOCOL SPECIFIC field (see table A0) contains a single numeric value as defined in 3.5.

Table A0 – SECURITY PROTOCOL SPECIFIC field for security protocol zzh

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

[Editors note: It is intended that the definition of a second SA Creation security protocol be accompanied by the addition of a Supported SA Creation 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 Creation protocols by modifying the data in the SCSI Cryptographic Algorithms IKE Payload (see 7.7.4.11).]

7.7.2.2 IKEv2-SCSI Device Server Capabilities step parameter data

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

Table ROW1 – IKEv2-SCSI Device Server Capabilities parameter data

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)	PARAMETER DATA LENGTH (n-3)						
3								(LSB)
4	SCSI SA Creation Capabilities payload (see 7.7.4.11)							
n								

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

The SCSI Cryptographic Algorithms payload (see 7.7.4.11) indicates the algorithms supported by the Key Exchange step (see 5.13.4.3) and Authentication step (see 5.13.4.4).

7.7.3 IKEv2-SCSI SECURITY PROTOCOL IN parameters

7.7.3.0a Overview

The purpose a SECURITY PROTCOL IN command in the IKEv2-SCSI protocol (i.e., when the SECURITY PROTOCOL field set to xxh) is to transfer SA creation and/or authentication related information from the device server. A SECURITY PROTOCOL IN command in which the SECURITY PROTOCOL field is set to xxh is associated with an previous SECURITY PROTOCOL OUT command as defined in 5.13.4.

If the IKEv2-SCSI SA creation protocol (see 5.13.4) is supported, the SECURITY PROTOCOL value of xxh shall be supported by the SECURITY PROTOCOL IN command as defined in this standard.

7.7.3.0b CDB description

When the SECURITY PROTOCOL field is set to IKEv2-SCSI (i.e., xxh) in a SECURITY PROTOCOL IN command, the SECURITY PROTOCOL SPECIFIC field (see table A1) contains a single numeric value as defined in 3.5.

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

Code	Description	Support	Reference
0000h – 00FFh	Restricted		RFC 4306
0100h – 0101h	Reserved		
0102h	Key Exchange step	Mandatory	5.13.4.3 and 7.7.3.1
0103h	Authentication step	Mandatory	5.13.4.4 and 7.7.3.1
0104h – EFFFh	Reserved		
F000h – FFFFh	Vendor Specific		

Any SECURITY PROTOCOL IN command with an allocation length of up to 16 384 bytes and the SECURITY PROTOCOL field is set to IKEv2-SCSI shall not be terminated with an error due to the number of bytes to be transferred.

7.7.3.1 IKEv2-SCSI parameter data format

7.7.3.1.1 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).

Table E1 – IKEv2-SCSI parameter data

Bit Byte	7	6	5	4	3	2	1	0
IKEv2-SCSI 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	EXCHANGE TYPE							
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 application client shall use this field to associate the parameter data in a SECURITY PROTOCOL IN COMMAND with the IKEv2-SCSI CCS in progress on the I_T_L nexus (see 5.13.4.6). If the application client cannot make this association, it shall abandon the CCS and shall not create an SA (see 5.13.4.6)

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 Key Exchange step 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 device server shall use this field to associate the Authentication step SECURITY PROTOCOL OUT command with the IKEv2-SCSI CCS in progress on the I_T_L nexus (see 5.13.4.6). If the device server cannot make this association, the command shall be terminated with CHECK CONDITION status, with the sense key set to NOT READY, and the additional sense code set to IKEv2-SCSI OPERATION IN PROGRESS, see 5.13.4.6.

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 EXCHANGE TYPE field shall be set to the step of the IKEv2-SCSI protocol:

a) F2h for the Key Exchange step (see 5.13.4.3); or

b) F3h for the Authentication step (see 5.13.4.4); or

c) F4h for a Delete operation (see 5.13.5); or

d) A value from the range F8h-FFh for a vendor-specific exchange.

NOTE – RFC 4306 specifies exchange types F0h-FFh as being for private use.

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 IKEv2-SCSI PARAMETER VALUE INVALID.

The VERSION bit shall be set to zero and ignored upon receipt (see RFC 4306). 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 RSPNS bit shall be set to zero for SECURITY PROTOCOL OUT commands and shall be set to one for SECURITY PROTOCOL IN commands.

The MESSAGE ID field contains an incrementing value that identifies a particular message (SECURITY PROTOCOL OUT command) and response (SECURITY PROTOCOL IN command) pair. The first MESSAGE ID in the Key Exchange step 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 and shall set the RSPNS bit to one. 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).

If the device server receives a SECURITY PROTOCOL OUT command with an invalid MESSAGE ID field in its parameter data, 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.

If the application client receives an invalid MESSAGE ID field in the parameter data for a SECURITY PROTOCOL IN command, the application client shall abandon the CCS and shall not create an SA (see 5.13.4.6).

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

Byte \ Bit	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								
...	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

[ROW Note: This needs to be agglomerated with the above subclause but I do not have the will to wrestle with Word to do it.]

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 [c]	IN Support	OUT Support	Reference
00h		No Next Payload	Mandatory	Mandatory	7.7.4.2
01h-20h			Reserved	Reserved	
21h	SA	Security Association [a]	Reserved	Reserved	RFC 4306
22h	KE	Key Exchange	Mandatory	Mandatory	7.7.4.3
23h	ID	Identification – Application Client	Reserved	Mandatory	7.7.4.4
24h	ID	Identification – Device Server	Mandatory	Reserved	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-IC [b]	Notify	Reserved	Mandatory	7.7.4.8a
2Ah	D	Delete	Reserved	Mandatory	7.7.4.8b
2Bh	V	Vendor ID	Mandatory	Mandatory	7.7.4.9
2Ch	TS	Traffic Selector – Application Client	Reserved	Reserved	RFC 4306
2Dh	TS	Traffic Selector – Device Server	Reserved	Reserved	RFC 4306
2Eh	E	Encrypted	Mandatory	Mandatory	7.7.4.10
2Fh	CP	Configuration	Reserved	Reserved	RFC 4306
30h	EAP	Extensible Authentication	Reserved	Reserved	RFC 4306
31h-7Fh			Restricted	Restricted	RFC 4306
80h	SSCC	SCSI SA Creation Capabilities	Mandatory	Reserved	7.7.4.11
81h	SCA	SCSI Cryptographic Algorithms	Mandatory	Mandatory	7.7.4.12
82h	STV	SCSI Timeout Values	Mandatory	Mandatory	7.7.4.13
83h-BFh			Reserved	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 reserved.

[C] RFC 4306 identifies the source of many payloads by appending a lowercase i or r to the name (e.g., KEi is a Key Exchange payload sent by the IKEv2 initiator). In IKEv2-SCSI, this identification is made based on the command being processed. Initiator payloads (e.g., KEi) always appear in the parameter data for a SECURITY PROTOCOL OUT command. Receiver payloads (e.g., KEr) always appear in the parameter data for a SECURITY PROTOCOL IN command. In some cases, different next payload coded values are used to distinguish RFC 4306 initiators and receivers. IKEv2-SCSI uses these values without changes but has no dependencies on them.

Payload lengths are not required to be multiples of 4 bytes, so payloads may not be aligned to 4 byte boundaries, see RFC 4718.

Some payloads use values defined for IKEv2 in registries maintained by IANA. In all cases, the values to be used are in the IANA Internet Key Exchange Version 2 (IKEv2) Parameters registry located at http://www.iana.org/assignments/ikev2-parameters .

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							
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 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. ID_DER_ASN1_DN shall be used when the identity is the value of a certificate subject field (see RFC 3280). ID_DER_ASN1_GN shall be used when the identity is the value of a name contained in a Subject Alternative Name (SubjectAltName) certificate extension (see RFC 3280. These ID Types 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.
c) ID_FC_NAME allows FC-SP certificates that certify a Fibre Channel name as an identity to be used, see RFC 4595 and FC-SP.

Other ID Types shall not be used.

When a certificate is used, the identity in the Identification payload is not required to match anything in the certificate, see RFC 4306, but it shall be possible to configure any application client or device server to require a match between the identity in an Identification payload and the subject name or subject alternative name in a certificate.

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. Table H2 shows the possible values of the CERTIFICATE ENCODING field in both payloads for IKEv2-SCSI.

Table H2 - CERTIFICATE ENCODING field values

Code	**Description**	**Reference**
00h	Reserved	
01h-03h	Restricted	RFC 4718
04h	X.509 Certificate - Signature	RFC 4306
05h-0Ah	Restricted	RFC 4718
0Bh	Raw RSA Key	RFC 4718
0Ch-0Dh	Restricted	7.7.4.5
0Eh-C8h	Reserved to IANA	RFC 4306
C9h-FFh	Reserved	7.7.4.5

In accordance with the recommendations in RFC 4718, certificate encoding values 01h-03h and 05h-0Ah shall not be used. This standard forbids Hash and URL certificate encodings, hence certificate encoding values 0Ch and 0Dh shall not be used. Certificate encoding values defined as vendor specific in RFC 4306 are reserved in this standard.

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 Digital Signature support is optional. Shared Key Message Integrity Code authentication shall be supported.
c) The device server prepends the contents of an SSCC payload to its Key Exchange step IKEv2-SCSI message in constructing the block of data to be signed. The SSCC payload shall be the SSCC payload that would be returned if a Device Capabilities step were performed at the time when the Authentication payload is constructed.
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).
e) An RSA digital signature shall be encoded with the EMSA-PKCS1-v1_5 signature encoding method as specified in RFC 2437, see RFC 4718.

The means for provisioning shared secrets for Shared Key Message Integrity Code authentication are outside the scope of this standard; these shared secrets may be provisioned at the time of manufacturing, during device or system initialization, or at any time thereafter. The following requirements from RFC 4306 apply to interfaces for provisioning shared secrets:

a) ASCII strings of at least 64 octets shall be supported.
b) A null terminator shall not be added to any input before using it as a shared secret.
c) A hexadecimal ASCII encoding of the shared secret shall be supported.
d) ASCII encodings other than hexadecimal may be supported. Support for any such encoding shall include specification of the algorithm for translating the encoding to a binary string as part of the interface.

Keys for the Authentication step of IKEv2-SCSI shall be generated as specified in RFC 4306. RFC 4718 contains a description of the octets to be signed; this description applies to IKEv2-SCSI with the following changes and clarifications:

a) The InitiatorSignedOctets are signed by the application client.
b) An SSCC payload is prepended to the ResponderSignedOctets as specified above. The result is signed by the device server.
c) GenIKEHDR does not apply. The "[four octets 0 if using port 4500]" do not exist in SCSI.
d) SPIi is the Application Client SAI, SPIr is the Device Server SAI.
e) RESERVED refers to a reserved field in the Identification payload (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.

The requirements that RFC 4306 places on nonces shall be followed for nonces used in IKEv2-SCSI.

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 ID payload (see 7.7.4.4) in the Key Exchange step SECURITY PROTOCOL OUT command (see 5.13.4.2). 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 H3 – Notify payload format

Bit Byte	7	6	5	4	3	2	1	0
0	NEXT PAYLOAD							
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 H4 – 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)
4	PROTOCOL ID							
5	SAI SIZE							
6	NUMBER OF SAIs (0001h)							
7								
8	(MSB)	APPLICATION CLIENT SECURITY ASSOCIATION INDEX						
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; this SAI shall be the same as the SAI in the IKE_SA APPLICATION CLIENT SAI field in the IKEv2-SCSI header of the SECURITY PROTOCOL OUT command that contains the Delete payload. The device server shall remove the corresponding security association from its own tables.

If an Delete payload is received with an APPLICATION CLIENT SECURITY ASSOCIATION INDEX field that is not the same as the SAI in the IKE_SA APPLICATION CLIENT SAI field in the IKEv2-SCSI header of the SECURITY PROTOCOL OUT command that contains the Delete payload, 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.

7.7.4.9 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.

The INTEGRITY CHECKSUM DATA in an Encrypted payload shall be checked by the recipient of the payload (application client for SECURITY PROTOCOL IN, device server for SECURITY PROTOCOL OUT). If the device server receives an Encrypted payload with invalid INTEGRITY CHECKSUM DATA, 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. If the application client receives an Encrypted payload with invalid INTEGRITY CHECKSUM DATA and a CCS is in progress, the application client shall abandon the CCS and shall not create an SA (see 5.13.4.6).

7.7.4.11 SCSI SA Creation Capabilities payload

An IKE payload type value of 80h indicates a SCSI Cryptographic Algorithms payload. For IKEv2-SCSI, this payload is used only in the IKEv2-SCSI Device Server Capabilities step parameter data (see 7.7.2.2)

Table I1 – SCSI SA Creation Capabilities 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	RESERVED							
5								
6								
7	NUMBER OF TRANSFORMS							
Algorithm Descriptors								
12	ALGORITHM DESCRIPTORS (VARIABLE)							
m								

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

The NEXT PAYLOAD field shall be set to zero.

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

The ALGORITHM DESCRIPTORS shall contain a SCSI Cryptographic Algorithm descriptor (see 7.7.4.11.1) for each algorithm that the device server is prepared to use with the application client that issued the SECURITY PROTOCOL IN command. There shall be at least one descriptor for each of the following algorithm types:

- Encryption Algorithm (ENCR)
- Pseudo-random Function (PRF)
- Integrity Algorithm (INTEG)
- Diffie-Hellman Group (D-H)
- IKE Authentication Algorithm (IKE-AUTH)

The SCSI Cryptographic Algorithms descriptors shall be ordered by:

1. Increasing ALGORITHM TYPE;
2. Increasing ALGORITHM IDENTIFIER within the same ALGORITHM TYPE; and
3. Increasing key length within the same ALGORITHM IDENTIFIER.

If the application client receives an SSCC payload that does not contain at least one descriptor for each required algorithm type, the application client shall not perform the Key Exchange step with the device server.

7.7.4.11.1 SCSI Cryptographic Algorithms descriptor overview

Each SCSI Cryptographic Algorithm descriptor (see table J1) specifies one algorithm used for authentication, integrity checking, or authentication.

Table J1 – SCSI Cryptographic Algorithm descriptor format

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).

Algorithm attribute values are defined in the subclauses that describe the algorithm types (see table K1).

Algorithm type values that RFC 4306 defines as vendor specific are reserved in this standard.
If the device server receives an SCA payload containing an unsupported algorithm 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. The sense data shall have the SKSV bit set to one and contain sense key specific sense data for the ILLEGAL REQUEST sense key in which the FIELD POINTER field designates the position of the first unsupported algorithm or key length.

[Editor's Note: IKEV2-SCSI PARAMETER VALUE INVALID is a new ASC/ASCQ]

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

In the IKEv2-SCSI Device Server Capabilities step (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 are allowed to be used in SA creation negotiations with the application client that issued the SECURITY PROTOCOL IN command. If an encryption algorithm is supported with more than one key length, one SCSI Cryptographic Algorithms descriptor shall be included for each key length.

The SCSI Cryptographic Algorithms descriptors shall be ordered by:

1. Increasing ALGORITHM TYPE;
2. Increasing ALGORITHM IDENTIFIER within the same ALGORITHM TYPE; and
3. Increasing key length within the same ALGORITHM IDENTIFIER.

Failure to observe this ordering may result in errors that are reported during the Authentication step (see 5.13.4.4) because the device server and application client do not agree on the data transferred by the SECURITY PROTOCOL IN command.

In the Key Exchange step SECURITY PROTOCOL OUT command (see 5.13.4.3.2) parameter data, this payload is used to specify the algorithms that the application client has selected. The device server echoes this payload to confirm acceptance of those algorithms in the Key Exchange step SECURITY PROTOCOL IN command (see 5.13.4.3) parameter data.

In the Key Exchange step, 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 NONE 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 indicated during the IKEv2-SCSI Device Server Capabilities step.

7.7.4.11.1.1 Encryption Algorithm (ENCR) identifiers

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

Table K2 – Encryption algorithm identifiers

Value	Description	Key Length (bytes)	Support	Reference
0000 000Bh	ENCR_NULL	0	Mandatory	RFC 2410
0000 000Ch	ENCR_AES_CBC	16 or 32 [a]	Mandatory	RFC 3602
0000 0010h	ENCR_AES_CCM_16 [b]	16 or 32 [a]	Optional	RFC 4309
0000 0014h	AES_GCM with a 16 octet ICV [c]	16 or 32 [a]	Optional	RFC 4106
0000 0400h – 0000 FFFFh	Vendor Specific	Vendor Specific		IANA
0001 0000h – FFFF FFFFh	Reserved			
All other values	Restricted			IANA

[a] 16 byte (128 bit) AES keys shall be supported, 32 byte (256 bit) AES keys may be supported, 24 byte (192 bit) AES keys shall not be used.

[b] AES CCM requires 3 bytes of keying material in addition to the AES key for use as a salt, see RFC 4309.

[c] AES GCM requires 4 bytes of keying material in addition to the AES key for use as a salt, see RFC 4106.

ENCR_NULL indicates that encryption is not to be used. ENCR_NULL may be used to omit encryption when integrity protection is required, but encryption is not required.

ENCR_AES_CCM_16 and AES_GCM are combined mode algorithms that provide both encryption and cryptographic integrity. The AUTH_NONE authentication algorithm shall be used with these combined mode algorithms.

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. The valid KEY LENGTH values depend on the encryption algorithm; the valid values for each algorithm are specified in Table K2. Support of more than one key length for an encryption algorithm is indicated in an SSCC payload by the presence of multiple encryption algorithm descriptors for that algorithm in the SCSI SA Creation Capabilities payload, one descriptor for each supported key length for that algorithm.

If the device server receives an SCA payload containing an invalid key length or a key length that the device server does not support, 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. The sense data shall have the SKSV bit set to one and contain sense key specific sense data for the ILLEGAL REQUEST sense key in which the FIELD POINTER field designates the position of the first unsupported algorithm or key length.

7.7.4.11.1.2 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
0000 0002h	PRF_HMAC_SHA1	Optional	FIPS 198a [a]
0000 0005h	PRF_HMAC_SHA2_256	Mandatory	RFC TBD
0000 0006h	PRF_HMAC_SHA2_384	Optional	RFC TBD
0000 0007h	PRF_HMAC_SHA2_512	Optional	RFC TBD
0400 – FFFFh	Vendor Specific		IANA
0001 0000h – FFFF FFFFh	Reserved		
All other values	Restricted		IANA

[a] The PRF is HMAC(K, text) for the SHA-1 hash function as specified in FIPS 198a.

The ALGORITHM ATTRIBUTES for PRF algorithms are RESERVED.

7.7.4.11.1.3 Integrity Algorithm (INTEG) identifiers

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

Table K4 – Integrity algorithm identifiers

Value	Description	Key Length (bytes)	Support	Reference
0000 0000h	AUTH_NONE	0	Optional	RFC 4306
0000 0002h	AUTH_HMAC_SHA1_96	20	Optional	RFC 2404
0000 000Ch	AUTH_HMAC_SHA2_256_128	32	Mandatory	RFC TBD
0000 000Dh	AUTH_HMAC_SHA2_384_192	48	Optional	RFC TBD
0000 000Eh	AUTH_HMAC_SHA2_512_256	64	Optional	RFC TBD
0000 0400h – 0000 FFFFh	Vendor Specific			IANA
0001 0000h – FFFF FFFFh	Reserved			
All other values	Restricted			IANA

AUTH_NONE indicates that no separate integrity algorithm is used; AUTH_NONE shall be supported when any combined mode (encryption and authentication) algorithm is supported (see 7.7.4.11.1.1). A device server shall report support for AUTH_NONE in the SCSI SA Creation Capabilities payload if it reports support for any combined mode algorithm. In the Key Exchange phase, an application client shall select the AUTH_NONE algorithm when it selects a combined mode algorithm. An application client shall not select AUTH_NONE when it selects the AES_CBC encryption algorithm or any other encryption algorithm that does not provide a cryptographic integrity check.

The key length used with an integrity algorithm is determined by the algorithm identifier (see table K4). The ALGORITHM ATTRIBUTES for INTEG algorithms are RESERVED.

7.7.4.11.1.4 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
0000 000Eh	2048-bit MODP group (finite field D-H)	256	Mandatory	RFC 3526
0000 000Fh	3072-bit MODP group (finite field D-H)	384	Optional	RFC 3526
0000 0010h	4096-bit MODP group (finite field D-H)	512	Optional	RFC 3526
0000 0011h	6144-bit MODP group (finite field D-H)	768	Optional	RFC 3526
0000 0012h	8192-bit MODP group (finite field D-H)	1024	Optional	RFC 3526
0000 0013h	256-bit prime elliptic curve field P-256	32	Optional	RFC 4753
0000 0014h	384-bit prime elliptic curve field P-384	48	Optional	RFC 4753
0000 0015h	521-bit prime elliptic curve field P-521	66	Optional	RFC 4753
0000 0400h – 0000 FFFFh	Vendor specific			IANA
0001 0000h – FFFF FFFFh	Reserved			
All other values	Restricted			IANA

The ALGORITHM ATTRIBUTES for D-H algorithms are RESERVED.

7.7.4.11.1.5 IKE Authentication Algorithm Type (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
0000 0000h	IKE_AUTH_NONE	Optional	7.7.4.11.1.5
0000 0001h	RSA Digital Signature [a]	Optional	RFC 4306
0000 0002h	Shared Key Message Integrity Code	Mandatory	RFC 4306
0000 0009h	ECDSA with SHA-256 on the P-256 curve [a]	Optional	RFC 4754
0000 000Ah	ECDSA with SHA-384 on the P-384 curve [a]	Optional	RFC 4754
0000 000Bh	ECDSA with SHA-512 on the P-521 curve [a]	Optional	RFC 4754
0000 00C9h – 0000 00FFh	Vendor Specific		IANA
0001 0100h – FFFF FFFFh	Reserved		
All other values	Restricted		IANA

[a] Use of certificates with this digital signature authentication algorithm is optional.

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, the IKEv2-SCSI Authentication step is skipped and the resulting SAs are not authenticated. The IKE_AUTH_NONE authentication algorithm shall not appear in the SA Device Capabilities parameter data except under the circumstances described in 5.13.4.3.4.

The Shared Key Message Integrity Code is based on a shared secret associated with the identity in the corresponding Identification payload (see 7.7.4.4).

The ALGORITHM ATTRIBUTES for IKE Authentication Algorithms are specified in Table K7.

Table K7 – IKE Authentication Algorithms - Attributes

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

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 Cryptographic Algorithms payload

An IKE payload type value of 81h indicates a SCSI Cryptographic Algorithms payload. This payload replaces the IKE Security Association payload (see RFC 4306).

Table L1 – 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	SECURITY ASSOCIATION TYPE							
5								
6	USAGE DATA LENGTH (k)							
7								
8	(MSB)	SAID						
15								(LSB)
16	(MSB)	USAGE DATA (VARIABLE)						
16+k-1								(LSB)
Algorithm Descriptors								
16+k	RESERVED							
16+k+1								
16+k+2								
16+k+3	NUMBER OF TRANSFORMS (5)							
16+k+4	ALGORITHM DESCRIPTORS (VARIABLE)							
m								

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

The NEXT PAYLOAD field shall be set to 22h (i.e., Key Exchange payload).

The SECURITY ASSOCIATION TYPE field shall be set to a value from table x2 in 5.13.2.

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 be specified by the command set referenced by the applicable row of table L2.

The SAID shall be set to the SAI of the entity that creates the payload. For a SECURITY PROTOCOL OUT command, the application client shall set the SAID field shall be set to the contents of the IKE_SA APPLICATION CLIENT SAI field in the IKEv2-SCSI header (see 7.7.3.1.1). For a SECURITY PROTOCOL IN command, the device server shall set the SAID field shall be set to the contents of the IKE_SA DEVICE SERVER SAI field in the IKEv2-SCSI header (see 7.7.3.1.1).

The USAGE DATA shall contain additional data that specifies how the created security association is to be used. The interpretation of USAGE DATA is specific to each value of SECURITY ASSOCIATION TYPE.

The NUMBER OF TRANSFORMS field shall be set to 5, as that is the number of algorithm descriptors in the payload.

The ALGORITHM DESCRIPTORS shall contain a SCSI Cryptographic Algorithm descriptor (see 7.7.4.11.1) for each algorithm that the device server is prepared to use with the application client that issued the SECURITY PROTOCOL IN command. There shall be one descriptor for each of the following algorithm types:

- Encryption Algorithm (ENCR)
- Pseudo-random Function (PRF)
- Integrity Algorithm (INTEG)
- Diffie-Hellman Group (D-H)
- IKE Authentication Algorithm (IKE-AUTH)

The SCSI Cryptographic Algorithms descriptors shall be ordered by increasing ALGORITHM TYPE.

If the device server receives an SCA payload that does not contain all the required descriptors, 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 PARAMETER LIST INCOMPLETE.

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

7.7.4.13 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	RESERVED							
5								
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 Key Exchange and Authentication steps (see 5.13.4.3 and

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.

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 to ILLEGAL REQUEST and the additional sense code set to IKEv2-SCSI PARAMETER VALUE INVALID. The device server shall enforce a maximum value for the IKEv2-SCSI PROTOCOL TIMEOUT and may enforce a maximum value for the IKEV2-SCSI SA INACTIVITY TIMEOUT. These maximum values are specified by means outside the scope of this standard. If an STV payload is received with a timeout that exceeds the applicable maximum 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 IKEv2-SCSI PARAMETER VALUE INVALID.

7.7.5 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
0000h	Reserved		
0001h	UNSUPPORTED_CRITICAL_PAYLOAD	ILLEGAL REQUEST	IKEv2-SCSI PARAMETER NOT SUPPORTED
0004h	INVALID_IKE_SPI	ILLEGAL REQUEST	IKEv2-SCSI PARAMETER VALUE INVALID
0005h	INVALID_MAJOR_VERSION	ILLEGAL REQUEST	IKEv2-SCSI PARAMETER VALUE INVALID
0007h	INVALID_SYNTAX [a]	ILLEGAL REQUEST	IKEv2-SCSI PARAMETER VALUE INVALID
0009h	INVALID_MESSAGE_ID	ILLEGAL REQUEST	IKEv2-SCSI PARAMETER VALUE INVALID
000Bh	INVALID_SPI	ILLEGAL REQUEST	IKEv2-SCSI PARAMETER VALUE INVALID [b]
000Eh	NO_PROPOSAL_CHOSEN [c]	ILLEGAL REQUEST	IKEv2-SCSI PARAMETER VALUE INVALID
0011h	INVALID_KE_PAYLOAD [c]	ILLEGAL REQUEST	IKEv2-SCSI PARAMETER VALUE INVALID
0018h	AUTHENTICATION_FAILED	ABORTED COMMAND	AUTHENTICATION FAILED
0022h - 0027h	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 step (see 5.13.4.2) and selects algorithms from them in the Key Exchange step (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 step to rediscover the device server's capabilities.

[d] These IKEv2 Error Types are used for features that are not supported by IKEv2-SCSI SA creation.

[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 a field in the command that caused the error.

7.7.6 IKEv2-SCSI SECURITY PROTOCOL OUT parameters

7.7.6.1 Overview

The purpose a SECURITY PROTCOL OUT command in the IKEv2-SCSI protocol (i.e., when the SECURITY PROTOCOL field set to xxh) is to transfer SA creation and/or authentication related information from the application client. A SECURITY PROTOCOL OUT command in which the SECURITY PROTOCOL field is set to xxh is associated with an previous SECURITY PROTOCOL IN command as defined in 5.13.4.

If the IKEv2-SCSI SA creation protocol (see 5.13.4) is supported, the SECURITY PROTOCOL value of xxh shall be supported by the SECURITY PROTOCOL OUT command as defined in this standard.

7.7.6.2 CDB description

When the SECURITY PROTOCOL field is set to IKEv2-SCSI (i.e., xxh) in a SECURITY PROTOCOL OUT command, the SECURITY PROTOCOL SPECIFIC field (see table D1) contains a single numeric value as defined in 3.5.

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

Code	Description	Support	Reference
0000h – 00FFh	Restricted		RFC 4306
0100h – 0101h	Reserved		
0102h	Key Exchange step	Mandatory	5.13.4.3 and 7.7.3.1
0103h	Authentication step	Mandatory	5.13.4.4 and 7.7.3.1
0104h	Delete	Mandatory	TBD
0105h – EFFFh	Reserved		
F000h – FFFFh	Vendor Specific		

Any SECURITY PROTOCOL OUT command with a transfer length of up to 16 384 bytes and the SECURITY PROTOCOL field is set to IKEv2-SCSI shall not be terminated with an error due to the number of bytes to be transferred.