Date: 18 May 2007
To: T10 Technical Committee
From: Matt Ball (Quantum) and David Black (EMC) — {{edited by Ralph O. Weber}}
Subject: SPC-4: Establishing a Security Association using IKEv2

## Introduction

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.

## References

| | |
|---|---|
| T10/SSC-3r3c | SCSI-3 Stream Commands. |
| T10/SPC-4r11 | 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) |

## 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; and
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.

## Revision History

r0 Initial revision with lots of help from Ralph Weber.
r1 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 ;-).]

r2 Incorporate comments from January Orlando meeting and additional design work. Major changes:
- Use separate IKEv2-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).

r3 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.

r4 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.

r5 Convert to FrameMaker and edit for T10 style

r6 Complete conversion to FrameMaker and incorporate comments from May CAP WG (minutes in 07-212), including the concept of tying the mandatory SA authentication algorithm to the USAGE_TYPE SA parameter (see 5.13.2.2).

Changes between r5 and r6 are so extensive that change bars would not be informative and so they are not used.

Unless otherwise indicated additions are shown in blue, deletions in ~~red strikethrough~~, and comments in green.

## Proposed Changes in SPC-4 r10

## Introduction

The SCSI Primary Commands - 4 (SPC-4) standard is divided into the following clauses and annexes:

| | |
|---|---|
| Clause 1 | is the scope. |
| Clause 2 | enumerates the normative references that apply to this standard. |
| Clause 3 | describes the definitions, symbols, and abbreviations used in this standard. |
| Clause 4 | describes the conceptual relationship between this document and the SCSI-3 Architecture Model. |
| Clause 5 | describes the command model for all SCSI devices. |
| Clause 6 | defines the commands that may be implemented by any SCSI device. |
| Clause 7 | defines the parameter data formats that may be implemented by any SCSI device. |
| Clause 8 | defines the well known logical units that may be implemented by any SCSI device. |
| Annex A | identifies differences between the terminology used in this standard and previous versions of this standard. (informative) |
| Annex B | describes the PERSISTENT RESERVE OUT command features necessary to replace the reserve/release management method and provides guidance on how to perform a third party reservation using persistent reservations. (informative) |
| Annex C | identifies the differences between IKEv2 (see RFC 4306) and the IKEv2-SCSI SA creation protocol defined by this standard. |
| Annex ~~C~~D | lists code values in numeric order. (informative) |
| Annex ~~D~~E | lists assigned vendor identifiers. (informative) |

…

## 2.2 Normative References

…

ISO/IEC 14165-251, *Fibre Channel Framing and Signaling Interface (FC-FS) [ANSI INCITS 373-2003]*

ISO/IEC 14165-431, *Fibre Channel Security Protocols (FC-SP) [ANSI INCITS 426-2007]*

IEC 60027:2000, *Letter symbols to be used in electrical technology - Part 2: Telecommunications and electronics*

…

## 2.5 IETF References

{{add the following references to those already listed and maintain RFC number order}}

RFC 2104, *HMAC: Keyed-Hashing for Message Authentication*

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 4434, *The AES-XCBC-PRF-128 Algorithm for the Internet Key Exchange Protocol (IKE)*

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)*

RFC 4868, *Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec*

…

## 3.1 Definitions

{{insert the following in the proper alphabetical order.}}

**3.1.c cryptographic command sequence (CCS):** A defined sequence of SECURITY PROTOCOL IN commands (see 6.29) and SECURITY PROTOCOL OUT commands (see 6.30) that realize the cryptographic protocol of a specified security operation (e.g., the SECURITY PROTOCOL IN commands and SECURITY PROTOCOL OUT commands in the Key Exchange step and Authentication step, if any, of an IKEv2-SCSI (see 3.1.f) SA (see 3.1.119) creation transaction).

**3.1.f IKEv2-SCSI:** Internet Key Exchange protocol version 2 for SCSI. See 5.13.4.

**3.1.g IKEv2-SCSI keys:** Secure keys (see 3.1.w) used to provide security for IKEv2-SCSI operations (e.g., creation and deletion of the SA). IKEv2-SCSI keys that are needed after SA creation is complete are maintained in the MGMT_DATA SA parameter (see 3.1.103). The secure keys names used by IKEv2-SCSI are listed in 5.13.4.4.

**3.1.h IKEv2-SCSI CCS:** The CCS (see 3.1.c) that is the Key Exchange step and Authentication step, if any, of an IKEv2-SCSI (see 3.1.f) SA (see 3.1.119) creation transaction.

**3.1.k Prohibited:** A keyword used to describe a feature, function, or coded value that is defined in a standard to which this standard makes a normative reference whose use is not allowed for implementations of this standard.

**3.1.o SA creation transaction:** Any sequence of SECURITY PROTOCOL IN commands (see 6.29) and SECURITY PROTOCOL OUT commands (see 6.30), including but not limited a CCS (see 3.1.c), that is used to create an SA between an application client and device server. See .

**3.1.p SA generation:** Computation and initialization of the SA parameter values (see 3.1.103) required to create an SA (see 3.1.119). This is the final step in creating an SA. It is performed separately by the application client and device server after all SCSI commands required to create an SA have been performed without error.

**3.1.q SA keys:** Secure keys (see 3.1.w) that are maintained in the USAGE_DATA SA parameter (see 3.1.103) and used to provide security for the operations that use the SA (e.g., encryption of command parameter data). The secure keys names used by IKEv2-SCSI are listed in 5.13.4.4.

**3.1.r SA participant:** An application client or device server that participates in the creation or use of an SA (see 3.1.119).

**3.1.w secure key (SK):** A cryptographically protected complex secret (i.e., not a password) that is known only to a defined and limited set of entities (e.g., one application client and one device server). Two kinds of secure keys are associated with SA (see 3.1.119) operation: IKEv2-SCSI keys (see 3.1.g) and SA keys (see 3.1.q). The secure keys names used by IKEv2-SCSI are listed in 5.13.4.4.

## 3.2 Symbols and acronyms

{{insert the following in the proper alphabetical order.}}

**CCS** cryptographic command sequence (see 3.1.c)
**PKI** Public Key Infrastructure (see RFC 3280)
**PRF** Pseudo-Random Function (see RFC 4306)
**SK** Secure Key (see 3.1.w)
**SK_ai** Secure Key for IKEv2-SCSI Data-Out Buffer Encrypted payload integrity checking (see 5.13.4)
**SK_ar** Secure Key for IKEv2-SCSI Data-In Buffer Encrypted payload integrity checking (see 5.13.4)
**SK_d** Secure Key for use by the requestor of an IKEv2-SCSI SA creation (see 5.13.4)
**SK_ei** Secure Key for IKEv2-SCSI Data-Out Buffer Encrypted payload encryption (see 5.13.4)
**SK_er** Secure Key for IKEv2-SCSI Data-In Buffer Encrypted payload encryption (see 5.13.4)
**SK_pi** Secure Key used to construct the IKEv2-SCSI Data-Out Buffer Authentication payload (see 5.13.4)
**SK_pr** Secure Key used to construct the IKEv2-SCSI Data-In Buffer Authentication payload (see 5.13.4)
…

### 5.6.3 Exceptions to SPC-2 RESERVE and RELEASE behavior

…

{{Insert the following new subclause in the Persistent Reservations model and renumber all subsequent subclauses.}}

### 5.6.4 Persistent reservations interactions with IKEv2-SCSI SA creation

If a PERSISTENT RESERVE OUT command is received while an IKEv2-SCSI CCS is in progress (see 5.13.4), the command shall be terminated with a CHECK CONDITION status, with the sense key NOT READY, and the additional sense code set to LOGICAL UNIT NOT READY, SA CREATION IN PROGRESS. The sense key specific additional sense data may be set as described in 5.13.7.

{{LOGICAL UNIT NOT READY, SA CREATION IN PROGRESS is a new additional sense code.}}

…

### 5.13.2 Security associations

#### 5.13.2.1 Principals of ~~security associations~~ SAs

…

### 5.13.2.2 SA parameters

… {{Only the last paragraph and table in 5.13.2.2 are modified by this proposal.}} …

The USAGE_TYPE SA parameter shall be one of the values shown in table 45. Table 45 also shows which IKEv2-SCSI SA_AUTH algorithm (see 7.7.3.6.6) is mandatory for each usage type.

**Table 45 — USAGE_TYPE SA parameter values**

| Value | Description | Mandatory SA_AUTH algorithm identifier (see table x31) | Reference |
|---|---|---|---|
| 0000h - 0080h | Reserved | | |
| 0081h | Tape Data Encryption | 009F 0000h | SSC-3 |
| 0082h - FFFFh | Reserved | | |

{{009F 0000h is SA_AUTH_NONE and the author expects this value to be replaced as the result of an agreement recorded in the minutes of the SSC-3 WG or CAP WG before the proposal is approved for incorporation in SPC-4.}}

### 5.13.2.3 Creating a security association

The SECURITY PROTOCOL IN command (see 6.29) and SECURITY PROTOCOL OUT command (see 6.30) security protocols shown in table 46 are used to create SAs. The process of creating an SA establishes the SA parameter (see 5.13.2.2) values as follows:

a) Initial values for:
   A) Both (i.e., application client and device server) sequence numbers set to zero; and
   B) All KEYMAT bytes set to zero;
b) Unchanging values for the lifetime of the SA:
   A) Both SAIs;
   B) TIMEOUT;
   C) Both nonces;
   D) KDF_ID;
   E) USAGE_TYPE;
   F) USAGE_DATA; and
   G) MGMT_DATA;
   and
c) Values that are zero upon completion of SA creation:
   A) KEY_SEED.

**Table 46 — Security protocols ~~that~~ used to create SAs**

| Security Protocol Code | Description | Reference |
|---|---|---|
| ~~TBD~~ | ~~TBD~~ | ~~TBD~~ |
| zzh | SA creation capabilities | 7.7.2 |
| xxh | IKEv2-SCSI | 5.13.4 |

### 5.13.3 Key derivation functions

…

### 5.13.4 Using IKEv2-SCSI to create a security association

{{All of 5.13.4, 5.13.5, 5.13.6, and 5.13.7 are new. Additions/deletions markups are not applied in these subclauses.}}

#### 5.13.4.1 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 3.1.119).

An IKEv2-SCSI SA creation transaction (see 3.1.o) shall only be initiated by the application client.

The IKEv2-SCSI protocol creates the following pair of IKE 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.

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

1) Device Server Capabilities step (see 5.13.4.6): The application client determines the device server's cryptographic capabilities;
2) Key Exchange step (see 5.13.4.7): The application client and device server:
   A) Perform a key exchange;
   B) Determine SAIs (see 3.1.120); and
   C) May complete the creation of the SA;
   and
3) Authentication step (see 5.13.4.8): 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 (see 6.29) and SECURITY PROTOCOL OUT command (see 6.30) identify the step for the IKEv2-SCSI protocol (see 7.7.3.2).

The Key Exchange step and the Authentication step depend on the results from the Device Capabilities step in order to create an SA. During the IKEv2-SCSI Key Exchange step, the application client and device server perform independent computations to construct the following sets of secure keys:

a) Secure keys that are used by the IKEv2-SCSI Authentication step; and
b) Secure keys that are used by the IKEv2-SCSI Authentication step and to delete the SA; and
c) Secure keys are used by SCSI operations that obtain security from the generated SA.

More details about these secure keys are provided in 5.13.4.4.

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 sending a sequence of two or four IKEv2-SCSI commands over a single I_T_L nexus to the device server. These commands constitute an IKEv2-SCSI CCS (see 3.1.h):

1) A Key Exchange step SECURITY PROTOCOL OUT command (see 5.13.4.7.2);
2) A Key Exchange step SECURITY PROTOCOL IN command (see 5.13.4.7.3);
3) An Authentication step SECURITY PROTOCOL OUT command (see 5.13.4.8.2); and
4) An Authentication step SECURITY PROTOCOL IN command (see 5.13.4.8.3).

The device server shall process each command in the IKEv2-SCSI CCS to completion before returning status. While a command in the IKEv2-SCSI CCS is being processed by the device server, the application client may use the REQUEST SENSE command (see 5.13.7) to ascertain the device server’s progress for the command.

If the application client sends the next command in the IKEv2-SCSI CSS before receiving status for the proceeding command, then the results are unpredictable.

The device server shall return the appropriate parameter data for any SECURITY PROTOCOL IN command in the IKEv2-SCSI CCS every time the command is received (i.e., receiving the same SECURITY PROTOCOL IN command multiple times shall not affect the device server’s view of IKEv2-SCSI CCS progress or correctness of operation).

The device server or application client may abandon the IKEv2-SCSI CCS before the SA is created (see 5.13.5).

The device server shall maintain state for the IKEv2-SCSI CCS from the time the Key Exchange step SECURITY PROTOCOL OUT command is completed with GOOD status until one of the following occurs:

a) The IKEv2-SCSI CCS completes successfully;
b) A command in the IKEv2-SCSI CCS is terminated with a status other than one of those shown in table x2 (see 5.13.5); or

c) The number of seconds specified in the IKEV2-SCSI PROTOCOL TIMEOUT field of the IKEv2-SCSI Timeout Values payload (see 7.7.3.5.14) in the Key Exchange step SECURITY PROTOCOL OUT parameter data elapses and none of the following commands have been received:
   A) The next command in the IKEv2-SCSI CCS; or
   B) A REQUEST SENSE command;
   or
d) One of the following event-related conditions (see SAM-4) occurs:
   A) Power cycle;
   B) Hard reset;
   C) Logical unit reset; or
   D) I_T nexus loss.

If the device server receives a SECURITY PROTOCOL OUT or SECURITY PROTOCOL IN command with the SECURITY PROTOCOL field set to IKEv2-SCSI (i.e., xxh) on an I_T_L nexus other than the one for which IKEv2-SCSI CCS state is being maintained, then the command shall be terminated with CHECK CONDITION status, with the sense key set to ABORTED COMMAND, and the additional sense code set to CONFLICTING SA CREATION REQUEST.

{{CONFLICTING SA CREATION REQUEST is a new additional sense code.}}

Except for the cases described in this subclause, the fact that the device server is maintaining IKEv2-SCSI CCS state on a particular I_T_L nexus shall not affect the processing of new commands received on that I_T_L nexus.

If the application client and device server agree to use SA_AUTH_NONE during the Key Exchange step (see 5.13.4.5), then:

a) The Authentication step is skipped;
b) The IKEv2-SCSI CCS consists of the two Key Exchange step commands;
c) SA creation occurs upon the completion of the Key Exchange step.

If either SA participant requires that the Authentication step be used, the device server shall not complete SA creation until successful completion of Authentication 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. The Authentication step should be performed if there is any doubt as to whether it is needed.

NOTE x1 - Omission of the Authentication step provides no defense against a man-in-the-middle adversary that is capable of modifying SCSI commands. 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, (e.g., 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 FC-SP).

### 5.13.4.2 IKEv2-SCSI Protocol summary

This subclause graphically summarizes the IKE-v2-SCSI payloads (see 7.7.3.5) that are exchanged between an application client and a device server during all steps of an IKEv2-SCSI SA creation transaction (see 3.1.o). Each IKEv2-SCSI step (see 5.13.4.1) is shown in a separate figure.

Figure x1 shows the Device Server Capabilities step (see 5.13.4.6). The Device Server Capabilities step consists of a SECURITY PROTOCOL IN command carrying an IKEv2-SCSI SA Creation Capabilities payload (see 7.7.3.5.12). The IKEv2-SCSI header is not used.

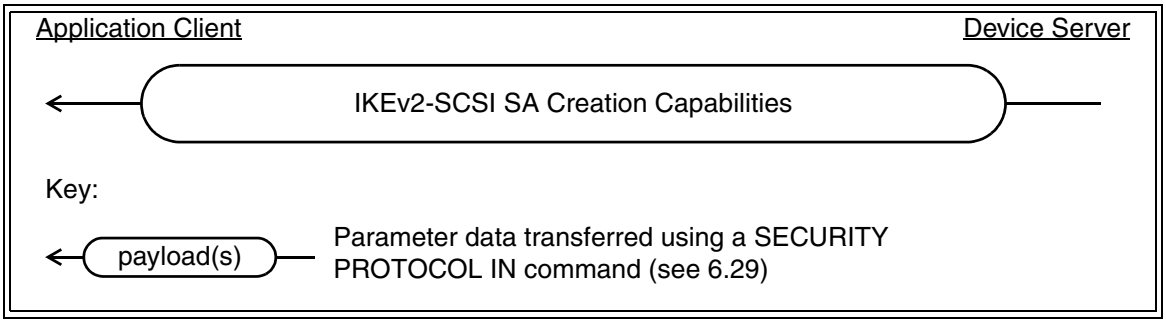


**Figure x1 — IKEv2-SCSI Device Server Capabilities step**

The IKEv2-SCSI SA Creation Capabilities payload indicates the device server's capabilities for SA creation.

Figure x2 shows the Key Exchange step (see 5.13.4.7). The Key Exchange step consists of a SECURITY PROTOCOL OUT command followed by a SECURITY PROTOCOL IN command.

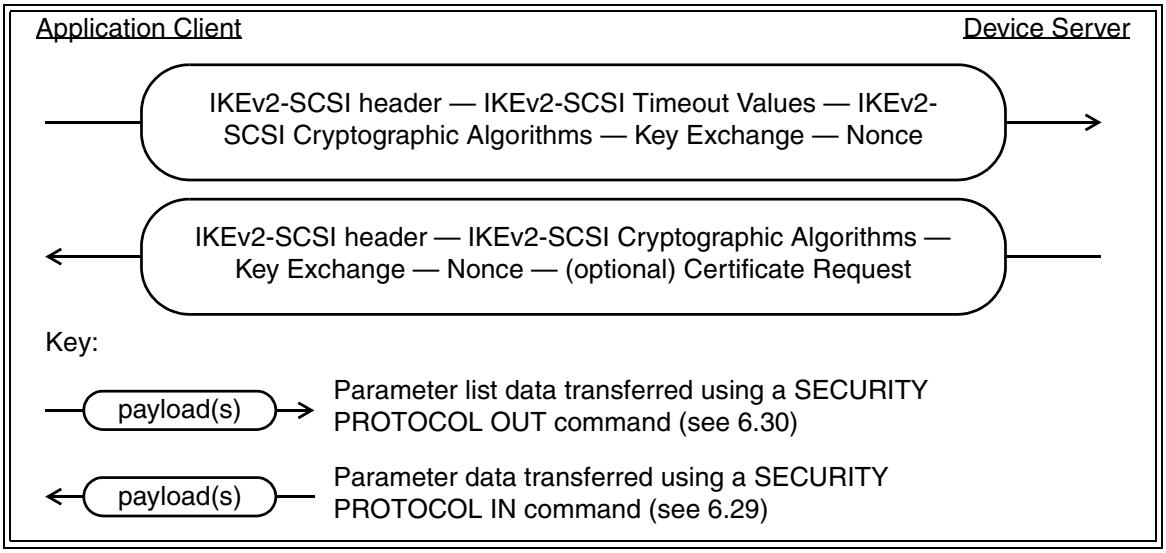


**Figure x2 — IKEv2-SCSI Key Exchange step**

The IKEv2-SCSI Timeout Values payload contains timeouts for SA creation and usage. The IKEv2-SCSI Cryptographic Algorithms payloads select and agree on usage of the SA and the cryptographic algorithms for the SA. The Key Exchange and Nonce payloads are part of the key and nonce exchanges that are used to generate SA keys. The optional Certificate Request payload enables the device server to request a certificate from the application client.

Figure x3 shows the Authentication step (see 5.13.4.8). The Authentication step consists of a SECURITY PROTOCOL OUT command followed by a SECURITY PROTOCOL IN command.

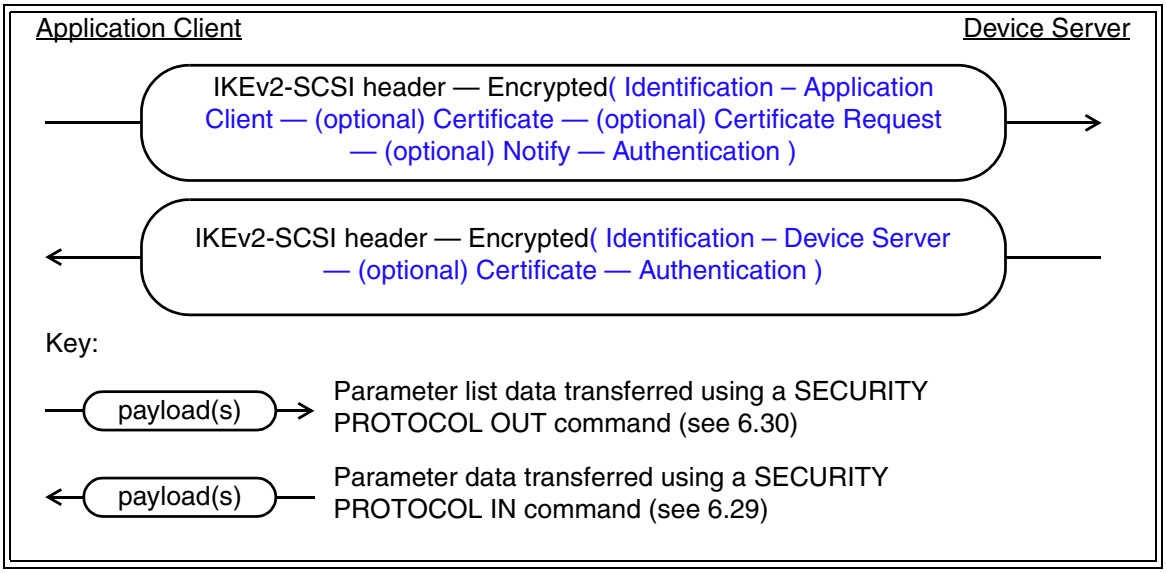


**Figure x3 — IKEv2-SCSI Authentication step**

{{The blue in figure x3 is intended for inclusion in SPC-4 and is used in conformance with the T10 Style Guide to assist viewers capable of rendering color to more easily see the encrypted payloads. It is not normative.}}

All payloads in the Authentication step are protected using the cryptographic algorithms determined by the IKEv2-SCSI Cryptographic Algorithms payloads in the Key Exchange step.

The Identification payloads contain the identities to be authenticated and are not required to be SCSI names or identities.

The optional Certificate Request payload allows an application client or device server to request the delivery of a Certificate payload in the parameter data for the next command (see 5.13.4.3).

The optional Notify payload provides a means to delete stale SAs between the same SA participants.

The Authenticate payloads authenticate not only the SA participants, but also the entire protocol sequence (e.g., the Authenticate payloads prevent a man-in-the-middle attack from succeeding).

#### 5.13.4.3 Handling of the Certificate Request payload and the Certificate payload

As detailed in this subclause, a Certificate Request payload (see 7.7.3.5.5) in one set of parameter data requests the delivery of a Certificate payload (see 7.7.3.5.5) in the next set of parameter data transferred. The purpose of this IKEv2-SCSI protocol construct is as follows:

a) Each SA participant is allowed to require the delivery of a Certificate payload by the other SA participant for use in authentication; and
b) Each Certificate Request payload indicates the trust anchors list used by the device server or application client when PKI-based Authentication is being used (see RFC 3280).

The presence of a Certificate Request payload in the Key Exchange step SECURITY PROTOCOL IN command (see 5.13.4.7.3) parameter data indicates that the device server requires the application client to send a Certificate payload in the Authentication step SECURITY PROTOCOL OUT command (see 5.13.4.8.2).

The presence of a Certificate Request payload in the Key Exchange step SECURITY PROTOCOL IN command parameter list specifies that the application client requires the device server to send a Certificate payload in the Authentication step SECURITY PROTOCOL IN command (see 5.13.4.8.3).

If any Certificate payloads are included in the parameter data, the first Certificate payload shall contain the public key used to verify the Authentication payload.

The application client and device server may use different authentication methods that require or do not require the use of Certificate payloads, and the presence or absence of Certificate Request payloads and Certificate payloads may vary in any of the commands described in this subclause.

#### 5.13.4.4 Summary of IKEv2-SCSI secure keys nomenclature

To facilitate use of the normative references made by this standard, the IKEv2-SCSI secure keys (see 3.1.w) are named as shown in table x1.

**Table x1 — IKEv2-SCSI secure key names**

| Name | Description | SA parameter (see 3.1.103) that stores this secure key |
|---|---|---|
| Secure keys used only during IKEv2-SCSI SA creation | | |
| SK_ar | The secure key used to integrity check the Encrypted payload (see 7.7.3.5.11) in the SECURITY PROTOCOL IN parameter data in the Authentication step (see 5.13.4.8.3). | Outside the scope of this standard |
| SK_er | The secure key used to encrypt the Encrypted payload in the SECURITY PROTOCOL IN parameter data in the Authentication step. | |
| SK_pi | The secure key used in the Authentication payload (see 7.7.3.5.6) construction for the SECURITY PROTOCOL OUT parameter list in the Authentication step (see 5.13.4.8.2). | |
| SK_pr | The secure key used in the Authentication payload construction for the SECURITY PROTOCOL IN parameter data in the Authentication step. | |
| Secure keys used during IKEv2-SCSI SA creation and deletion | | |
| SK_ai | The secure key used to integrity check the Encrypted payload in the SECURITY PROTOCOL OUT parameter list in the:<br>a) Authentication step; and<br>b) IKEv2-SCSI Delete command (see 5.13.6). | MGMT_DATA |
| SK_ei | The secure key used to encrypt the Encrypted payload in the SECURITY PROTOCOL OUT parameter list in the:<br>a) Authentication step; and<br>b) IKEv2-SCSI Delete command. | |
| Secure keys used only after the IKEv2-SCSI SA has been created | | |
| SK_d | The secure key used to encrypt or integrity check data processed under the protection of the IKEv2-SCSI SA. | KEY_SEED |

#### 5.13.4.5 Skipping the Authentication step

{{The requirements in the following paragraph deserve CAP study. They are not unheard of in SCSI, just uncommon.}}

In the Device Server Capabilities step (see 5.13.4.6) the parameter data returned by the SECURITY PROTOCOL IN command (see 7.7.2.3.1) in the IKEv2-SCSI SA Creation Algorithms payload (see 7.7.3.5.12) in an SA_AUTH IKEv2-SCSI cryptographic algorithm descriptor (see 7.7.3.6.6), an ALGORITHM IDENTIFIER field set to SA_AUTH_NONE (see 7.7.3.6.6) indicates that the device server permits the Authentication step to be omitted (see 5.13.4.1).

The device server shall not return SA_AUTH_NONE as an Authentication payload authentication algorithm identifier in the Device Server Capabilities step unless the device server has been configured to do so by a person who represents the owner of the SCSI target device that contains the device server. The methods for configuring a device server to return SA_AUTH_NONE are outside the scope of this standard. Device servers shall not be manufactured to return SA_AUTH_NONE as an Authentication payload authentication algorithm type in the Device Server Capabilities step.

In the Key Exchange step SECURITY PROTOCOL OUT command (see 5.13.4.7.2), the application client requests that the Authentication step be omitted by setting the ALGORITHM IDENTIFIER field to SA_AUTH_NONE in the SA_AUTH cryptographic algorithm descriptor in the IKEv2-SCSI Cryptographic Algorithms payload (see 7.7.3.5.13).

The application client should not select the SA_AUTH_NONE value as an Authentication payload authentication algorithm type unless:

a) An an SA_AUTH IKEv2-SCSI cryptographic algorithm descriptor from the Device Server Capabilities step indicates SA_AUTH_NONE availability; and
b) The application client is configured to omit the Authentication step by an administrator.

NOTE x2 - When SA_AUTH_NONE is used, IKEv2-SCSI has no protection against any man-in-the-middle attacks. Enabling return of the SA_AUTH_NONE authentication algorithm type in the Device Capabilities step, and allowing an application client to select SA_AUTH_NONE in the Key Exchange step are 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. 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 such as FC-SP).

#### 5.13.4.6 Device Server Capabilities step

In the Device Server Capabilities step, the application client sends a SECURITY PROTOCOL IN command (see 6.29) with the SECURITY PROTOCOL field set to SA creation capabilities (i.e., zzh) and the SECURITY PROTOCOL SPECIFIC field set to 0101h.

The device server returns the SECURITY PROTOCOL IN parameter data specified by the SECURITY PROTOCOL SPECIFIC field (see 7.7.2.2) and the parameter data (see 7.7.2.3.1) contains an IKEv2-SCSI SA Creation Capabilities payload (see 7.7.3.5.12).

The Device Server Capabilities step participates in the negotiation to skip the Authentication step as described in 5.13.4.5.

NOTE x3 - 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.

#### 5.13.4.7 IKEv2-SCSI Key Exchange step

##### 5.13.4.7.1 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 5.13.4.7.2);
2) A SECURITY PROTOCOL IN command (see 5.13.4.7.3); and
3) Key exchange completion (see 5.13.4.7.4)

NOTE x4 - 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.

##### 5.13.4.7.2 Key Exchange step SECURITY PROTOCOL OUT command

To send its key exchange message to the device server, the application client sends a SECURITY PROTOCOL OUT command (see 6.30) with the SECURITY PROTOCOL field set to IKEv2-SCSI (i.e., xxh) and the SECURITY PROTOCOL SPECIFIC field set to 0102h. The parameter list consists of an IKEv2-SCSI header (see 7.7.3.4) and the following:

1) A IKEv2-SCSI Timeout Values payload (see 7.7.3.5.14);
2) A IKEv2-SCSI Cryptographic Algorithms payload (see 7.7.3.5.13);
3) A Key Exchange payload (see 7.7.3.5.3); and
4) A Nonce payload (see 7.7.3.5.7).

The IKEv2-SCSI Timeout Values payload contains the inactivity timeouts that apply to this IKEv2-SCSI SA creation transaction (see 3.1.o) and the SA (see 3.1.119) that is created.

The IKEv2-SCSI Cryptographic Algorithms payload contains the following information about the SA to be created:

a) The cryptographic algorithms selected by the application client; and
b) The usage data (see 7.7.3.5.13) that is specific to the SA.

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 should not perform the Key Exchange step to request the creation of an SA.

IKEv2-SCSI Cryptographic Algorithms payload error checking requirements that ensure a successful negotiation of SA creation algorithms are described in 7.7.3.6.

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

The Nonce payload contains the application client's random nonce (see 3.1.95).

##### 5.13.4.7.3 Key Exchange step SECURITY PROTOCOL IN command

If the Key Exchange step SECURITY PROTOCOL OUT command (see 5.13.4.7.2) completes with GOOD status, the application client sends a SECURITY PROTOCOL IN command (see 6.29) with the SECURITY PROTOCOL field set to IKEv2-SCSI (i.e., xxh) and 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 in response to the SECURITY PROTOCOL IN command shall contain an IKEv2-SCSI header (see 7.7.3.4) and the following:

1) A IKEv2-SCSI Cryptographic Algorithms payload (see 7.7.3.5.13);
2) A Key Exchange payload (see 7.7.3.5.3);
3) A Nonce payload (see 7.7.3.5.7); and
4) Zero or more Certificate Request payloads (see 5.13.4.3).

As part of 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. If the device server is unable to establish this association:
   A) The SECURITY PROTOCOL IN 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, SA CREATION IN PROGRESS; and
   B) The device server shall continue the IKEv2-SCSI CCS by preparing to receive another Key Exchange step SECURITY PROTOCOL IN command;
b) Return the device server's SAI in the IKEv2-SCSI header IKE_SA DEVICE SERVER SAI field;
c) Return the IKEv2-SCSI Cryptographic Algorithms payload containing:
   A) The cryptographic algorithms supplied by the application client in the Key Exchange step SECURITY PROTOCOL OUT command parameter list; and
   B) The device server's SAI (see 3.1.120) in the SAID field (see 7.7.3.5.13);
d) Return information about the completed the Diffie-Hellman exchange with the Key Exchange payload; and
e) Return device server's random nonce (see 3.1.95) in the Nonce payload.

{{LOGICAL UNIT NOT READY, SA CREATION IN PROGRESS is a new additional sense code.}}

If the Key Exchange step SECURITY PROTOCOL IN command (see 5.13.4.7.3) completes with GOOD status, the application client should copy the device server's SAI from the SAID field in the IKEv2-SCSI Cryptographic Algorithms payload to the state it is maintaining for the IKEv2-SCSI CCS.

The application client may ignore the other fields in the IKEv2-SCSI Cryptographic Algorithms payload, however a greater degree of procedural integrity checking is achieve if the application client compares the contents of the IKEv2-SCSI Cryptographic Algorithms payload to what was sent in the Key Exchange step SECURITY PROTOCOL OUT command (see 5.13.4.7.2). If the application client detects differences in the contents of the two IKEv2-SCSI Cryptographic Algorithms payloads other than in the SAID field, the application client should notify the device server that the IKEv2-SCSI CCS is being abandoned as described in 5.13.5.

**5.13.4.7.4 Key Exchange step completion**

Before completing the Key Exchange step SECURITY PROTOCOL IN command (see 5.13.4.7.3) with GOOD status the device server shall complete the Key Exchange step as described in this subclause.

Upon receipt of GOOD status for the Key Exchange step SECURITY PROTOCOL IN command the application client should complete the Key Exchange step as described in this subclause.

The SA participants complete the Key Exchange step as follows:

1) Generate SKEYSEED (see RFC 4306) using the PRF selected by the PRF IKEv2-SCSI cryptographic algorithm descriptor (see 7.7.3.6.3) in the IKEv2-SCSI Cryptographic Algorithms payload; and
2) Use SKEYSEED and the KDF selected by the PRF IKEv2-SCSI cryptographic algorithm descriptor to generate the all IKEv2-SCSI secure keys (see 5.13.4.4) without regard for whether or not they are needed in the following order:

1) SK_d;
2) SK_ai;
3) SK_ar;
4) SK_ei;
5) SK_er;
6) SK_pi; and
7) SK_pr.

{{The extra bits needed by the CCM and GCM algorithms are not yet covered in the above list.}}

If the IKEv2-SCSI Cryptographic Algorithms payload SA_AUTH IKEv2-SCSI cryptographic algorithm descriptor (see 7.7.3.6.6) contains SA_AUTH_NONE in the ALGORITHM IDENTIFIER field and the Key Exchange step does not end with the IKEv2-SCSI CCS being abandoned (see 5.13.5), the Authentication step is not performed (see 5.13.4.5). In this case, the SA participants generate the SA as defined in 5.13.4.9.

NOTE x5 - The SKEYSEED described in this subclause is different from the SKEYSEED SA parameter (see 3.1.103).

#### 5.13.4.8 IKEv2-SCSI Authentication step

##### 5.13.4.8.1 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 created SA.

The Authentication step is accomplished as follows:

1) A SECURITY PROTOCOL OUT command (see 5.13.4.8.2); and
2) A SECURITY PROTOCOL IN command (see 5.13.4.8.3).

The parameter data for both commands shall be encrypted and integrity protected using the algorithms and keys determined in the Key Exchange step (see 5.13.4.7.4).

NOTE x6 - The Authentication step corresponds to the IKEv2 IKE_AUTH exchange in RFC 4306.

##### 5.13.4.8.2 Authentication step SECURITY PROTOCOL OUT command

To send its authentication information to the device server, the application client sends a SECURITY PROTOCOL OUT command (see 6.30) with the SECURITY PROTOCOL field set to IKEv2-SCSI (i.e., xxh) and the SECURITY PROTOCOL SPECIFIC field set to 0103h. The parameter data consists of the IKEv2-SCSI header (see 7.7.3.4) and an Encrypted payload (see 7.7.3.5.11) that:

a) Is integrity checked using the following:
   A) The algorithm specified by the INTEG IKEv2-SCSI algorithm descriptor (see 7.7.3.6.4) in the IKEv2-SCSI Cryptographic Algorithms payload (see 7.7.3.5.13); and
   B) The SK_ai secure key (see 5.13.4.4);
b) Is Encrypted using the following:
   A) The algorithm specified by the ENCR IKEv2-SCSI algorithm descriptor (see 7.7.3.6.2) in the IKEv2-SCSI Cryptographic Algorithms payload; and
   B) The SK_ei (see 5.13.4.4);
   and

c) Contains the following:
   1) An Identification – Application Client payload (see 7.7.3.5.4);
   2) Zero or more Certificate payloads (see 5.13.4.3);
   3) Zero or more Certificate Request payloads (see 5.13.4.3);
   4) Zero or one Notify payload (see 7.7.3.5.8); and
   5) An Authentication payload (see 7.7.3.5.6).

Before performing any checks of data contained in the Encrypted payload, the device server validate the SECURITY PROTOCOL OUT command parameter data as follows and not abandon the IKEv2-SCSI CCS if any of the validation tests fail, then:

a) The device server shall compare the IKE_SA APPLICATION CLIENT SAI field and the IKE_SA APPLICATION CLIENT SAI field to the SAI values it is maintaining for the IKEv2-SCSI CCS as described in 7.7.3.4; and
b) The device server shall decrypt and check the integrity of the Encrypted payload as described in 7.7.3.5.11.

In the SECURITY PROTOCOL OUT command parameter list, the application client:

a) sends its identity with the ID payload;
b) sends knowledge of the secret corresponding to ID; and
c) integrity protects the prior steps using the Authentication payload.

The application client uses the Notify payload to send an initial contact notification to the device server. The initial contact notification specifies that the application client has no stored state for any SAs with the device server other than the SA that is being created.

In response to receipt of an initial contact notification, the device server should delete all other SAs that were authenticated with a SECURITY PROTOCOL OUT command that contained the same Identification - Application Client payload data as that which is present in the SECURITY PROTOCOL OUT command that the device server is processing.

If the device server deletes other SAs in response to an initial contact notification, it shall do so only after the successful completion of the Authentication step (see 5.13.4.8). If an error occurs during the Authentication step, the device server shall ignore the initial contact notification.

If the device server is unable to proceed with SA creation for any reason (e.g., the verification of the Authentication 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 Authentication payload fails, or when authentication fails for any other reason.

{{AUTHENTICATION FAILED is a new additional sense code}}

#### 5.13.4.8.3 Authentication step SECURITY PROTOCOL IN command

To obtain the device server's authentication information, the application client then sends a SECURITY PROTOCOL IN command (see 6.29) with the SECURITY PROTOCOL field set to IKEv2-SCSI (i.e., xxh) and the SECURITY PROTOCOL SPECIFIC field set to 0103h. The parameter data consists of the IKEv2-SCSI header (see 7.7.3.4) and an Encrypted payload (see 7.7.3.5.11) that:

a) Is integrity checked using the following:
   A) The algorithm specified by the INTEG IKEv2-SCSI algorithm descriptor (see 7.7.3.6.4) in the IKEv2-SCSI Cryptographic Algorithms payload (see 7.7.3.5.13); and
   B) The SK_ar secure key (see 5.13.4.4);

b) Is Encrypted using the following:
   A) The algorithm specified by the ENCR IKEv2-SCSI algorithm descriptor (see 7.7.3.6.2) in the IKEv2-SCSI Cryptographic Algorithms payload; and
   B) The SK_er (see 5.13.4.4);
   and
c) Contains the following:
   1) An Identification – Device Server payload (see 7.7.3.5.4);
   2) Zero or more Certificate payloads (see 5.13.4.3); and
   3) An Authentication payload (see 7.7.3.5.6).

In the SECURITY PROTOCOL IN parameter data, the device server:

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

Before returning GOOD status for the SECURITY PROTOCOL IN command, the device server shall generate the SA as described in 5.13.4.9.

The application client should verify the Authentication payload as described in 7.7.3.5.6. The Certificate payload(s) are used as part of this verification for PKI-based authentication. If the Authentication payload is verified and no other error occurs the application client should generate the SA as described in 5.13.4.9.

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 should:

a) Not use the SA for any additional activities; and
b) Notify the device server that the IKEv2-SCSI CCS is being abandoned as described in 5.13.5.

**5.13.4.9 SA generation**

The application client and the device server shall initialize the SA parameters (see 3.1.103) as follows:

a) AC_SAI shall be set to the value in the IKE_SA APPLICATION CLIENT SAI field in the IKEv2-SCSI header (see 7.7.3.4) in the Key Exchange step SECURITY PROTOCOL OUT command (see 5.13.4.7.2);
b) DC_SAI shall be set to the value in the IKE_SA DEVICE SERVER SAI field in the IKEv2-SCSI header in the Key Exchange step SECURITY PROTOCOL IN command (see 5.13.4.7.3);
c) TIMEOUT shall be set to the IKEV2-SCSI SA INACTIVITY TIMEOUT field in the IKEv2-SCSI Timeout Values payload (see 7.7.3.5.14) in the Key Exchange step SECURITY PROTOCOL OUT command;
d) AC_SQN shall be set to one;
e) DC_SQN shall be set to one;
f) AC_NONCE shall be set to the value of the NONCE DATA field in the Nonce payload (see 7.7.3.5.7) in the Key Exchange step SECURITY PROTOCOL OUT command;
g) 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;
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);
i) KDF_ID shall be set as described in 5.13.4.7.4;
j) KEYMAT should be set to zero;
k) USAGE_TYPE shall be set to the value in the SA TYPE field in the IKEv2-SCSI Cryptographic Algorithms payload (see 7.7.3.5.13) in the Key Exchange step SECURITY PROTOCOL OUT command;
l) USAGE_DATA shall contain at least the following values from of the usage data field in the IKEv2-SCSI Cryptographic Algorithms payload in the Key Exchange step SECURITY PROTOCOL OUT command:
   A) The USAGE DATA field;

B) The ALGORITHM IDENTIFIER field (see 7.7.3.6) in the IKEv2-SCSI Cryptographic Algorithm descriptor (see 7.7.3.6) for the ENCR algorithm type;
C) The KEY LENGTH field (see 7.7.3.6.2) in the ALGORITHM ATTRIBUTES field in the IKEv2-SCSI Cryptographic Algorithm descriptor for the ENCR algorithm type; and
D) The ALGORITHM IDENTIFIER field (see 7.7.3.6) in the IKEv2-SCSI Cryptographic Algorithm descriptor for the INTEG algorithm type;
and

m) MGMT_DATA shall contain at least the following values:
A) From the IKEv2-SCSI Cryptographic Algorithms payload (see 7.7.3.5.13) in the Key Exchange step SECURITY PROTOCOL OUT command:
a) The ALGORITHM IDENTIFIER field (see 7.7.3.6) in the IKEv2-SCSI Cryptographic Algorithm descriptor (see 7.7.3.6) for the ENCR algorithm type;
b) The KEY LENGTH field (see 7.7.3.6.2) in the ALGORITHM ATTRIBUTES field in the IKEv2-SCSI Cryptographic Algorithm descriptor for the ENCR algorithm type;
c) The ALGORITHM IDENTIFIER field (see 7.7.3.6) in the IKEv2-SCSI Cryptographic Algorithm descriptor for the INTEG algorithm type;
B) From the Key Exchange step completion (see 5.13.4.3.4);
a) The value of SK_ei; and
b) The value of SK_ai;
and
C) The next value of the message id field in the IKEv2-SCSI header.

{{The extra bits needed by the CCM and GCM algorithms are not yet covered in the MGMT_DATA portion of above list.}}

NOTE x7 - The inclusion of the algorithm identifiers and key length in USAGE_DATA SA parameter enables the SA to apply the same encryption and integrity algorithms that IKEv2-SCSI negotiated to future IKEv2-SCSI SECURITY PROTOCOL OUT commands.

### 5.13.5 Abandoning an IKEv2-SCSI CCS

The occurrence of errors in either the application client or the device server may require that an IKEv2-SCSI CCS (see 3.1.c) be abandoned.

A device server shall indicate that it has abandoned an IKEv2-SCSI CCS by terminating an IKEv2-SCSI CCS command (see 5.13.4.1) received on the I_T_L nexus for which the IKEv2-SCSI CCS state is being maintained with any combination of status and sense data other than those shown in table x2.

**Table x2 — IKEv2-SCSI command terminations that do not abandon the CCS**

| IKEv2-SCSI CCS Command | Status (Sense Key) | Additional Sense Code | Description |
|---|---|---|---|
| SECURITY PROTOCOL OUT<br>SECURITY PROTOCOL IN | GOOD (n/a) | n/a | Indicates IKEv2-SCSI CCS is progressing normally |
| Key Exchange step SECURITY PROTOCOL OUT (see 5.13.4.7.2) | CHECK CONDITION (ABORTED COMMAND) | CONFLICTING SA CREATION REQUEST | An IKEv2-SCSI CCS is already active, and attempts to start another are blocked until the first CCS completes |
| SECURITY PROTOCOL OUT<br>SECURITY PROTOCOL IN | CHECK CONDITION (NOT READY) | LOGICAL UNIT NOT READY, SA CREATION IN PROGRESS | Device server is busy processing another command in the IKEv2-SCSI CCS or a different IKEv2-SCSI CCS |
| SECURITY PROTOCOL IN | CHECK CONDITION (ILLEGAL REQUEST) | INVALID FIELD IN CDB | Incorrect SECURITY PROTOCOL IN CDB format |
| SECURITY PROTOCOL OUT | | UNABLE TO DECRYPT PARAMETER LIST | Authentication step SECURITY PROTOCOL OUT command (see 5.13.4.8.2) for which the device server is unable to decrypt the Encrypted payload (see 7.7.3.5.11) or the integrity check fails |

{{CONFLICTING SA CREATION REQUEST, LOGICAL UNIT NOT READY, SA CREATION IN PROGRESS and UNABLE TO DECRYPT PARAMETER LIST are a new additional sense codes.}}

As part of abandoning an IKEv2-SCSI CCS, the device server shall:

a) Discard all maintained state (see 5.13.4.1); and
b) Prepare to allow the next Key Exchange step SECURITY PROTOCOL OUT command received to start a new IKEv2-SCSI CCS.

After a device server abandons an IKEv2-SCSI CCS, the device server shall respond to all new IKEv2-SCSI protocol commits as if an IKEv2-SCSI CCS had never been started.

An application client should not abandon an IKEv2-SCSI CCS when the next command in the CCS is a SECURITY PROTOCOL IN command. Instead, the application client should send the appropriate SECURITY PROTOCOL IN command and then abandon the IKEv2-SCSI CCS.

An application client should specify that it has abandoned an IKEv2-SCSI CSS by:

1) Sending an IKEv2-SCSI Delete command (see 5.13.6), and

2) If GOOD status is not returned in response to the IKEv2-SCSI Delete command, sending 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 and the INTTR bit set to zero in the IKEv2-SCSI header (see 7.7.3.4).

### 5.13.6 Deleting an IKEv2-SCSI SA

When an SA is deleted, both sets of SA parameters (see 5.13.2.2) are deleted as follows:

1) The application client uses the information in its SA parameters to prepare an IKEv2-SCSI Delete command that requests deletion of the device server's SA parameters;
2) The application client deletes its SA parameters and any associated data;
3) The application client sends the IKEv2-SCSI Delete command prepared in step 1) to the device server;
4) In response to the IKEv2-SCSI Delete command, the device server deletes its SA parameters and any associated data.

The IKEv2-SCSI Delete command is 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 consists of the IKEv2-SCSI header (see 7.7.3.4) and an Encrypted payload (see 7.7.3.5.11) that contains the one Delete payload (see ).

The Delete payload should conform to the requirements described in .

The Encrypted payload shall be encrypted and integrity checked using the MGMT_DATA SA parameter for the SA that is being deleted.

If the device server is able to valid the integrity checking information and decrypt the Encrypted payload, and locate the specified SA, it shall delete its SA parameters and any associated data. If the device server is unable to process the SECURITY PROTOCOL OUT command parameter list, the command shall be terminated with a CHECK CONDITION status, with the sense key and additional sense code set as described in 7.7.3.7.

### 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, SA CREATION IN PROGRESS. The sense data should include sense key specific data for the NOT READY sense key (i.e., a PROGRESS INDICATION field indicating the progress of the device server in performing the necessary cryptographic calculations).

{{LOGICAL UNIT NOT READY, SA CREATION IN PROGRESS is a new additional sense code.}}

The device server shall not use the progress indication to report the detailed 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 detailed progress of the computation (e.g., divide a constant expected time for the computation by 10 and advance the progress indication in 10% increments based solely on the time).

The requirements 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.

{{New model subclause text ends here. Additions/deletions markups resume.}}

### 5.13.8~~5.13.4~~ Security algorithm codes

Table 51 lists the security algorithm codes used in security protocol parameter data.

**Table 51 — Security algorithm codes**

| Code [a] | Description | Reference |
|---|---|---|
| Encryption algorithms | | |
| 0001 000Bh | ENCR_NULL | 7.7.3.6.2 |
| 0001 000Ch | AES-CBC | RFC 3602 |
| 0001 0010h~~–[a]~~ | AES-CCM with a 16 byte MAC | NIST SP 800-38C |
| 0001 0014h~~–[a]~~ | AES-GCM with a 16 byte MAC | NIST SP 800-38D |
| 0000 0400h - 0000 FFFFh | Vendor specific | |
| PRF and KDF ~~Algorithms~~ algorithms | | |
| 0002 0002h~~–[a]~~ | IKEv2-use ~~based iterative HMAC KDF~~ based on SHA-1 | ~~5.13.3.3~~ ~~5.13.3.4~~ ~~5.13.3.3~~ ~~5.13.3.3~~ ~~5.13.3.3~~ table x25 (see 7.7.3.6.3) |
| 0002 0004h~~–[a]~~ | IKEv2-use ~~based iterative HMAC KDF~~ based on AES-128 in CBC mode | |
| 0002 0005h~~–[a]~~ | IKEv2-use ~~based iterative HMAC KDF~~ based on SHA-256 | |
| 0002 0006h~~–[a]~~ | IKEv2-use ~~based iterative HMAC KDF~~ based on SHA-384 | |
| 0002 0007h~~–[a]~~ | IKEv2-use ~~based iterative HMAC KDF~~ based on SHA-512 | |
| 0002 0400h - 0002 FFFFh | Vendor specific | |
| Integrity checking (i.e., AUTH) algorithms | | |
| 0003 0000h | AUTH_NONE | RFC 4306 |
| 0003 0002h | AUTH_HMAC_SHA1_96 | RFC 2404 |
| 0003 000Ch | AUTH_HMAC_SHA2_256_128 | RFC 4868 |
| 0003 000Dh | AUTH_HMAC_SHA2_384_192 | RFC 4868 |
| 0003 000Eh | AUTH_HMAC_SHA2_512_256 | RFC 4868 |
| F003 0000h | AUTH_COMBINED | 7.7.3.6.4 |
| 0003 0400h - 0003 FFFFh | Vendor specific | |

[a] The lower order 16 bits of ~~this~~ these code values are assigned to match an IANA assigned value, if any, for an equivalent IKEv2 encryption algorithm (see 3.1.54) and the high order 16 bits match the IANA assigned IKEv2 transform type (e.g.~~i.e.~~, 1 – Encryption Algorithms, 2 – PRFs ~~Pseudo-random Functions~~).

**Table 51 — Security algorithm codes**

| Code [a] | Description | Reference |
|---|---|---|
| Diffie-Hellman algorithms | | |
| 0004 000Eh | 2 048-bit MODP group (finite field D-H) | RFC 3526 |
| 0004 000Fh | 3 072-bit MODP group (finite field D-H) | RFC 3526 |
| 0004 0010h | 4 096-bit MODP group (finite field D-H) | RFC 3526 |
| 0004 0013h | 256-bit prime elliptic curve field P-256 | RFC 4753 |
| 0004 0015h | 521-bit prime elliptic curve field P-521 | RFC 4753 |
| 0004 0400h - 0004 FFFFh | Vendor specific | |
| SA Authentication payload authentication algorithms | | |
| 00F9 0000h | SA_AUTH_NONE | 7.7.3.6.6 |
| 00F9 0001h | RSA Digital Signature | RFC 4306 |
| 00F9 0002h | Shared Key Message Integrity Code | RFC 4306 |
| 00F9 0009h | ECDSA with SHA-256 on the P-256 curve | RFC 4754 |
| 00F9 000Bh | ECDSA with SHA-512 on the P-521 curve | RFC 4754 |
| 00F9 0400h - 00F9 FFFFh | Vendor specific | |
| Other algorithms | | |
| 0000 0000h – 0000 FFFFh | Restricted | IANA |
| ~~0000 0400h – 0000 FFFFh~~ | ~~Vendor specific~~ | |
| All others ~~value~~ | Reserved | |

[a] The lower order 16 bits of ~~this~~ these code values are assigned to match an IANA assigned value, if any, for an equivalent IKEv2 encryption algorithm (see 3.1.54) and the high order 16 bits match the IANA assigned IKEv2 transform type (e.g.~~i.e.~~, 1 – Encryption Algorithms, 2 – PRFs ~~Pseudo-random Functions~~).

…

## 6.29 SECURITY PROTOCOL IN command

### ~~6.29.1 SECURITY PROTOCOL IN command description~~

The SECURITY PROTOCOL IN command (see table 193) is used to retrieve security protocol information (see 6.29.2) or the results of one or more SECURITY PROTOCOL OUT commands (see 6.30).

**Table 193 — SECURITY PROTOCOL IN command**

**{{no changes in table 193 contents}}**

The SECURITY PROTOCOL field (see table 194) specifies which security protocol is being used.

**Table 194 — SECURITY PROTOCOL field in SECURITY PROTOCOL IN command**

| Code | Description | Reference |
|---|---|---|
| 00h | Security protocol information | ~~6.29.2~~ 7.7.1 |
| 01h - 06h | Defined by the TCG | 3.1.140 |
| 07h - 1Fh | Reserved | |
| 20h | Tape Data Encryption | SSC-3 |
| 21h | Data Encryption Configuration | TBD |
| {{insert two new rows (suggest 40h and 41h) and adjust reserved values accordingly}} | | |
| zzh | SA Creation Capabilities | 7.7.2 |
| xxh | IKEv2-SCSI | 7.7.3 |
| 22h - EDh | Reserved | |
| EEh | Authentication in Host Attachments of Transient Storage Devices | IEEE 1667 |
| EFh | ATA Device Server Password Security | TBD |
| F0h - FFh | Vendor Specific | |

Editors Note 1 - ROW: SECURITY PROTOCOL field code values 21h – 2Fh are tentatively reserved for SSC-x uses.

The contents of the SECURITY PROTOCOL SPECIFIC field depend on the protocol specified by the SECURITY PROTOCOL field (see table 194).

A 512 increment (INC_512) bit set to one specifies that the ALLOCATION LENGTH field (see 4.3.4.6) expresses the maximum number of bytes available to receive data in increments of 512 bytes (e.g., a value of one means 512 bytes, two means 1 024 bytes, etc.). Pad bytes may or may not be appended to meet this length. Pad bytes shall have a value of 00h. An INC_512 bit set to zero specifies that the ALLOCATION LENGTH field expresses the number of bytes to be transferred.

Indications of data overrun or underrun and the mechanism, if any, for processing retries depend on the protocol specified by the SECURITY PROTOCOL field (see table 194).

Any association between a previous SECURITY PROTOCOL OUT command and the data transferred by a SECURITY PROTOCOL IN command depends on the protocol specified by the SECURITY PROTOCOL field (see table 194). If the device server has no data to transfer (e.g., the results for any previous SECURITY PROTOCOL OUT commands are not yet available), the device server may transfer data indicating it has no other data to transfer.

The format of the data transferred depends on the protocol specified by the SECURITY PROTOCOL field (see table 194).

The device server shall retain data resulting from a SECURITY PROTOCOL OUT command, if any, until one of the following events is processed:

a) Transfer of the data via a SECURITY PROTOCOL IN command from the same I_T_L nexus as defined by the protocol specified by the SECURITY PROTOCOL field (see table 194);

b) Logical unit reset (See SAM-4); or
c) I_T nexus loss (See SAM-4) associated with the I_T nexus that sent the SECURITY PROTOCOL OUT command.

If the data is lost due to one of these events the application client may send a new SECURITY PROTOCOL OUT command to retry the operation.

~~**6.29.2 Security protocol information description**~~

~~**6.29.2.1 Overview**~~

~~...~~ {{Move the entire contents of 6.29.2 to 7.7.1.}} ~~...~~

~~**6.29.2.4.3 Attribute certificate description**~~

~~RFC 3281 defines the certificate syntax for certificates consistent with X.509v2 Attribute Certificate Specification. Any further restrictions beyond the requirements of RFC 3281 are yet to be defined by T10.~~

## 6.30 SECURITY PROTOCOL OUT command

The SECURITY PROTOCOL OUT command (see table 198) is used to send data to the logical unit. The data sent specifies one or more operations to be performed by the logical unit. The format and function of the operations depends on the contents of the SECURITY PROTOCOL field (see table 199). Depending on the protocol specified by the SECURITY PROTOCOL field, the application client may use the SECURITY PROTOCOL IN command (see 6.29) to retrieve data derived from these operations.

**Table 198 — SECURITY PROTOCOL OUT command**
**{{no changes in table 198 contents}}**

The SECURITY PROTOCOL field (see table 199) specifies which security protocol is being used.

**Table 199 — SECURITY PROTOCOL field in SECURITY PROTOCOL OUT command**

| Code | Description | Reference |
|---|---|---|
| 00h | Reserved | |
| 01h - 06h | Defined by the TCG | 3.1.140 |
| 07h - 1Fh | Reserved | |
| 20h | Tape Data Encryption | SSC-3 |
| 21h | Data Encryption Configuration | TBD |
| | {{insert one new row (suggest 41h) and adjust reserved values accordingly}} | |
| xxh | IKEv2-SCSI | 7.7.3 |
| 22h - EDh | Reserved | |
| EEh | Authentication in Host Attachments of Transient Storage Devices | IEEE 1667 |
| EFh | ATA Device Server Password Security | TBD |
| F0h - FFh | Vendor Specific | |

Editors Note 2 - ROW: SECURITY PROTOCOL field code values 21h – 2Fh are tentatively reserved for SSC-x uses.

The contents of the SECURITY PROTOCOL SPECIFIC field depend on the protocol specified by the SECURITY PROTOCOL field (see table 199).

A 512 increment (INC_512) bit set to one specifies that the TRANSFER LENGTH field (see 4.3.4.4) expresses the number of bytes to be transferred in increments of 512 bytes (e.g., a value of one means 512 bytes, two means 1 024 bytes, etc.). Pad bytes shall be appended as needed to meet this requirement. Pad bytes shall have a value of 00h. A INC_512 bit set to zero specifies that the TRANSFER LENGTH field indicates the number of bytes to be transferred.

Any association between a SECURITY PROTOCOL OUT command and a subsequent SECURITY PROTOCOL IN command depends on the protocol specified by the SECURITY PROTOCOL field (see table 199). Each protocol shall define whether:

a) The device server shall complete the command with GOOD status as soon as it determines the data has been correctly received. An indication that the data has been processed is obtained by sending a SECURITY PROTOCOL IN command and receiving the results in the associated data transfer; or
b) The device server shall complete the command with GOOD status only after the data has been successfully processed and an associated SECURITY PROTOCOL IN command is not required.

The format of the data transferred depends on the protocol specified by the SECURITY PROTOCOL field (see table 199).

…

## 7.7 Security protocol parameters

### 7.7.1 Security protocol information description

{{Move the entire contents of 6.29.2 here}}

### 7.7.2 SA creation capabilities

{{All text from here to the end of this proposal is new. Additions/deletions markups are not applied in these subclauses.}}

#### 7.7.2.1 Overview

If the SECURITY PROTOCOL field in a SECURITY PROTOCOL IN command (see 6.29) is set to zzh, then the command returns information related to the SA creation (see 5.13.2.3) capabilities provided by the device server.

The SA creation capabilities protocol is independent of any other SA creation protocols. The device server shall process an SA creation capabilities SECURITY PROTOCOL IN command at any time and this processing shall not affect the:

a) State maintained for any SA creation CCS (e.g., an IKEv2-SCSI CCS); or
b) Concurrent processing of any commands that are part of an SA creation CCS.

If any SA creation protocols are supported, the SA creation capabilities protocol shall be supported as described in 7.7.2.

The SA creation capabilities SECURITY PROTOCOL IN CDB format is described in 7.7.2.2.

As shown in table x3 (see 7.7.2.2), the format of the parameter data returned by a SA creation capabilities SECURITY PROTOCOL IN command depends on the value in the SECURITY PROTOCOL SPECIFIC field in the CDB.

### 7.7.2.2 SA creation capabilities CDB description

The SA creation capabilities SECURITY PROTOCOL IN CDB has the format defined in 6.29 with the additional requirements described in this subclause.

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 x3) identifies the SA creation protocol (see 5.13.2.3) for which the device server shall return capability information.

**Table x3 — SECURITY PROTOCOL SPECIFIC field for the SA creation capabilities SECURITY PROTOCOL IN command**

| Code | Description | Parameter data format |
|---|---|---|
| 0000h – 0100h | Reserved | |
| 0101h | IKEv2-SCSI device server capabilities | 7.7.2.3.1 |
| 0102h – EFFFh | Reserved | |
| F000h – FFFFh | Vendor Specific | |

{{It is intended that the definition of a second SA Creation protocol be accompanied by the addition of a security protocol specific code (probably 0000h or 0102h). However, it is also possible to introduce new SA Creation protocols by modifying the data in the IKEv2-SCSI Cryptographic Algorithms IKE Payload (see 7.7.3.5.13).}}

If an SA creation capabilities SECURITY PROTOCOL IN command is received with the INC_512 bit is set to one, then the SECURITY PROTOCOL IN command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

Any SA creation capabilities SECURITY PROTOCOL IN command with an allocation length of up to 16 384 bytes shall not be terminated with an error due to the number of bytes to be transferred and processed.

#### 7.7.2.3 SA creation capabilities parameter data formats

##### 7.7.2.3.1 IKEv2-SCSI device server capabilities parameter data format

The IKEv2-SCSI device server capabilities parameter data (see table x4) indicates the IKEv2 transforms (i.e., key exchange protocols and authentication protocols) supported by the device server for IKEv2-SCSI.

**Table x4 — IKEv2-SCSI device server capabilities parameter data**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | (MSB) | | | | | | | |
| 3 | PARAMETER DATA LENGTH (n-3) | | | | | | | (LSB) |
| 4 | IKEv2-SCSI SA Creation Capabilities payload (see 7.7.3.5.12) | | | | | | | |
| n | | | | | | | | |

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

The IKEv2-SCSI SA Creation Capabilities payload (see 7.7.3.5.12) indicates the algorithms supported by IKEv2-SCSI in the Key Exchange step (see 5.13.4.7) and Authentication step (see 5.13.4.8).

NOTE x8 - The primary content of the IKEv2-SCSI device server capabilities SA creation capabilities parameter data is an IKEv2-SCSI payload (see 7.7.3.5) because the IKEv2-SCSI SA Creation Capabilities payload is used by the device server and application client in the construction of the Authentication payload (see 7.7.3.5.6).

### 7.7.3 IKEv2-SCSI

#### 7.7.3.1 Overview

If the SECURITY PROTOCOL field in a SECURITY PROTOCOL OUT command (see 6.30) or a SECURITY PROTOCOL IN command (see 6.29) is set to xxh, then the command is part of an IKEv2-SCSI CCS (see 5.13.4) and is used to transfer IKEv2-SCSI protocol information to or from the device server.

In an IKEv2-SCSI CCS, a defined sequence of SECURITY PROTOCOL OUT and SECURITY PROTOCOL IN commands are sent by the application client and processed by the device server as summarized in 5.13.4.1.

The IKEv2-SCSI SECURITY PROTOCOL OUT CDB format is described in 7.7.3.3.

The IKEv2-SCSI SECURITY PROTOCOL IN CDB format is described in 7.7.3.2.

The IKEv2-SCSI SECURITY PROTOCOL OUT command and the IKEv2-SCSI SECURITY PROTOCOL IN command use the same parameter data format and this format is described in 7.7.3.4. A significant component of the IKEv2-SCSI parameter data format is one or more IKE payloads and the format of IKE payloads is described in 7.7.3.5.

If the IKEv2-SCSI SA creation protocol is supported (see 7.7.1), the SA creation capabilities protocol (see 7.7.2) shall also be supported.

### 7.7.3.2 IKEv2-SCSI SECURITY PROTOCOL IN CDB description

The IKEv2-SCSI SECURITY PROTOCOL IN CDB has the format defined in 6.29 with the additional requirements described in this subclause.

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 x5) identifies the IKEv2-SCSI step (see 5.13.4.1) that the device server is to process. If the IKEv2-SCSI SA creation protocol is supported (see 7.7.1), the SECURITY PROTOCOL IN command support requirements are shown in table x5.

**Table x5 — SECURITY PROTOCOL SPECIFIC field for the IKEv2-SCSI SECURITY PROTOCOL IN command**

| Code | Description | Support | Reference | |
|---|---|---|---|---|
| | | | Usage | Data format |
| 0000h – 00FFh | Restricted | | RFC 4306 | RFC 4306 |
| 0100h – 0101h | Reserved | | | |
| 0102h | Key Exchange step | Mandatory | 5.13.4.7.3 | 7.7.3.4 |
| 0103h | Authentication step | Mandatory | 5.13.4.8.3 | 7.7.3.4 |
| 0104h – EFFFh | Reserved | | | |
| F000h – FFFFh | Vendor Specific | | | |

If an IKEv2-SCSI SECURITY PROTOCOL IN command is received with the INC_512 bit is set to one while the device server is maintaining state for an IKEv2-SCSI CCS (see 5.13.4.1), then:

a) The SECURITY PROTOCOL IN 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, SA CREATION IN PROGRESS; and
b) The device server shall continue the IKEv2-SCSI CCS by preparing to receive another SECURITY PROTOCOL IN command or SECURITY PROTOCOL OUT command, as appropriate.

{{LOGICAL UNIT NOT READY, SA CREATION IN PROGRESS is a new additional sense code.}}

If an IKEv2-SCSI SECURITY PROTOCOL IN command is received with the INC_512 bit is set to one while the device server is not maintaining state for an IKEv2-SCSI CCS (see 5.13.4.1), then the SECURITY PROTOCOL IN command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

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

#### 7.7.3.3 IKEv2-SCSI SECURITY PROTOCOL OUT CDB description

The IKEv2-SCSI SECURITY PROTOCOL OUT CDB has the format defined in 6.30 with the additional requirements described in this subclause.

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 x6) identifies the IKEv2-SCSI step (see 5.13.4.1) that the device server is to process. If the IKEv2-SCSI SA creation protocol is supported (see 7.7.1), the SECURITY PROTOCOL IN command support requirements are shown in table x6.

**Table x6 — SECURITY PROTOCOL SPECIFIC field for the IKEv2-SCSI SECURITY PROTOCOL OUT command**

| Code | Description | Support | Reference | |
|---|---|---|---|---|
| | | | Usage | Data format |
| 0000h – 00FFh | Restricted | | RFC 4306 | RFC 4306 |
| 0100h – 0101h | Reserved | | | |
| 0102h | Key Exchange step | Mandatory | 5.13.4.7.3 | 7.7.3.4 |
| 0103h | Authentication step | Mandatory | 5.13.4.8.3 | 7.7.3.4 |
| 0104h | Delete operation | Mandatory | 5.13.6 | 7.7.3.4 |
| 0105h – EFFFh | Reserved | | | |
| F000h – FFFFh | Vendor Specific | | | |

If an IKEv2-SCSI SECURITY PROTOCOL OUT command is received with the INC_512 bit is set to one while the device server is maintaining state for an IKEv2-SCSI CCS (see 5.13.4.1), then:

a) The SECURITY PROTOCOL OUT 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, SA CREATION IN PROGRESS; and
b) The device server shall continue the IKEv2-SCSI CCS by preparing to receive another SECURITY PROTOCOL OUT command or SECURITY PROTOCOL IN command, as appropriate.

{{LOGICAL UNIT NOT READY, SA CREATION IN PROGRESS is a new additional sense code.}}

If an IKEv2-SCSI SECURITY PROTOCOL OUT command is received with the INC_512 bit is set to one while the device server is not maintaining state for an IKEv2-SCSI CCS (see 5.13.4.1), then 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 INVALID FIELD IN CDB.

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

#### 7.7.3.4 IKEv2-SCSI parameter data format

Table x7 shows the parameter list format used a SECURITY PROTOCOL OUT command and the parameter data format used by a SECURITY PROTOCOL IN command when the SECURITY PROTOCOL field is set to IKEv2-SCSI (i.e., xxh).

**Table x7 — IKEv2-SCSI SECURITY PROTOCOL OUT and SECURITY PROTOCOL IN 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 | | | |
| 18 | EXCHANGE TYPE | | | | | | | |
| 19 | Reserved | | | INTTR | VERSION | RSPNS | Reserved | |
| 20 | (MSB) | | | MESSAGE ID | | | | |
| 23 | | | | | | | | (LSB) |
| 24 | (MSB) | | | IKE LENGTH (n+1) | | | | |
| 27 | | | | | | | | (LSB) |
| IKEv2-SCSI payloads | | | | | | | | |
| 28 | | | | First IKEv2-SCSI payload (see 7.7.3.5) | | | | |
| | | | | . . . | | | | |
| n | | | | Last IKEv2-SCSI payload (see 7.7.3.5) | | | | |

The IKE_SA APPLICATION CLIENT SAI field contains the value that will become the AC_SAI SA parameter (see 5.13.2.2) when the SA is generated (see 5.13.4.9). The AC_SAI is chosen by the application client to uniquely identify its representation of the SA that is being negotiated.

If the device server receives an IKEv2-SCSI header with the IKE_SA APPLICATION CLIENT SAI field set to zero, then:

a) The command shall be terminated with a CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to SA CREATION PARAMETER VALUE INVALID; and
b) The device server shall abandon the IKEv2-SCSI CCS (see 5.13.5).

To increase procedural integrity checking, the application client should compare the IKE_SA APPLICATION CLIENT SAI field contents in any SECURITY PROTOCOL IN parameter data it receives to the value that the application client is maintaining for the IKEv2-SCSI CCS. If the two values are not identical, the application client should notify the device server that the IKEv2-SCSI CCS is being abandoned as described in 5.13.5.

Except in the Key Exchange step SECURITY PROTOCOL OUT command (see 5.13.4.7.2), the IKE_SA DEVICE SERVER SAI field contains the value that will become the DC_SAI SA parameter when the SA is generated. The DC_SAI is chosen by the device server in accordance with the requirements in 5.13.2.1 to uniquely identify its representation of the SA that is being negotiated. In the Key Exchange step SECURITY PROTOCOL OUT command the IKE_SA DEVICE SERVER SAI field is reserved.

To increase procedural integrity checking, the application client should compare the IKE_SA DEVICE SERVER SAI field contents in the Authentication step SECURITY PROTOCOL IN parameter data it receives to the value that the application client is maintaining for the IKEv2-SCSI CCS. If the two values are not identical, the application client should notify the device server that the IKEv2-SCSI CCS is being abandoned as described in 5.13.5.

The device server shall compare the contents of the IKE_SA APPLICATION CLIENT SAI field and the IKE_SA DEVICE SERVER SAI field in the Authentication step SECURITY PROTOCOL OUT parameter list to the SAI values the device server is maintaining for the IKEv2-SCSI CCS. If the values do not match, then:

a) The SECURITY PROTOCOL OUT 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, SA CREATION IN PROGRESS; and
b) The device server shall continue the IKEv2-SCSI CCS by preparing to receive another Authentication step SECURITY PROTOCOL OUT command.

The NEXT PAYLOAD field (see table x8) identifies the first IKEv2-SCSI payload that follows the IKEv2-SCSI header.

**Table x8 — NEXT PAYLOAD field**

| Code | IKE Payload Name | Support requirements in SECURITY PROTOCOL … | | Reference |
|---|---|---|---|---|
| | | IN | OUT | |
| 00h | No Next Payload | Mandatory | | 7.7.3.5.2 |
| 01h - 20h | | Reserved | | |
| 21h | Security Association | Reserved [a] | | RFC 4306 |
| 22h | Key Exchange | Mandatory | | 7.7.3.5.3 |
| 23h | Identification – Application Client | Reserved | Mandatory | 7.7.3.5.4 |
| 24h | Identification – Device Server | Mandatory | Reserved | 7.7.3.5.4 |
| 25h | Certificate | Optional | | 7.7.3.5.5 |
| 26h | Certificate Request | Optional | | 7.7.3.5.5 |
| 27h | Authentication | Mandatory | | 7.7.3.5.6 |
| 28h | Nonce | Mandatory | | 7.7.3.5.7 |
| 29h | Notify [b] | Reserved | Mandatory | 7.7.3.5.8 |
| 2Ah | Delete | Reserved | Mandatory | |
| 2Bh | Vendor ID | Mandatory | | 7.7.3.5.10 |
| 2Ch | Traffic Selector – Application Client | Reserved | | RFC 4306 |

[a] The Security Association payload type value is not used in IKEv2-SCSI. The IKEv2-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.

**Table x8 — NEXT PAYLOAD field**

| Code | IKE Payload Name | Support requirements in SECURITY PROTOCOL … | | Reference |
|---|---|---|---|---|
| | | IN | OUT | |
| 2Dh | Traffic Selector – Device Server | Reserved | | RFC 4306 |
| 2Eh | Encrypted | Mandatory | | 7.7.3.5.11 |
| 2Fh | Configuration | Reserved | | RFC 4306 |
| 30h | Extensible Authentication | Reserved | | RFC 4306 |
| 31h - 7Fh | | Restricted | | RFC 4306 |
| 80h | IKEv2-SCSI SA Creation Capabilities | Mandatory | | 7.7.3.5.12 |
| 81h | IKEv2-SCSI Cryptographic Algorithms | Mandatory | | 7.7.3.5.13 |
| 82h | IKEv2-SCSI Timeout Values | Mandatory | | 7.7.3.5.14 |
| 83h - BFh | | Reserved | | |
| C0h - FFh | Vendor Specific | | | |

[a] The Security Association payload type value is not used in IKEv2-SCSI. The IKEv2-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.

The MAJOR VERSION field shall contain the value 2h. If a device server receives an IKE header with a MAJOR VERSION field containing a value other than 2h, then:

a) The command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST and the additional sense code set to SA CREATION PARAMETER VALUE INVALID; and
b) The device server shall abandon the IKEv2-SCSI CCS (see 5.13.5).

The MINOR VERSION field is reserved.

The EXCHANGE TYPE field is reserved.

The initiator (INTTR) bit shall be set to:

a) One for SECURITY PROTOCOL OUT commands; and
b) Zero for SECURITY PROTOCOL IN commands.

If a device server receives an IKEv2-SCSI header with the INTTR bit set to zero, then:

a) The command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST and the additional sense code set to SA CREATION PARAMETER VALUE INVALID; and
b) The device server shall abandon the IKEv2-SCSI CCS (see 5.13.5).

If an application client receives an IKEv2-SCSI header with the INTTR bit set to one, it should notify the device server that the IKEv2-SCSI CCS is being abandoned as described in 5.13.5.

The VERSION bit is reserved.

The response (RSPNS) bit shall be set to:

a) Zero for SECURITY PROTOCOL OUT commands; and
b) One for SECURITY PROTOCOL IN commands.

If a device server receives an IKEv2-SCSI header with the RSPNS bit set to one, then:

a) The command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST and the additional sense code set to SA CREATION PARAMETER VALUE INVALID; and
b) The device server shall abandon the IKEv2-SCSI CCS (see 5.13.5).

If an application client receives an IKEv2-SCSI header with the RSPNS bit set to zero, it should notify the device server that the IKEv2-SCSI CCS is being abandoned as described in 5.13.5.

The MESSAGE ID field contains:

a) Zero for the Key Exchange step SECURITY PROTOCOL OUT command and SECURITY PROTOCOL IN command; and
b) One for the Authentication step SECURITY PROTOCOL OUT command and SECURITY PROTOCOL IN command.

If the device server receives a SECURITY PROTOCOL OUT command with an invalid MESSAGE ID field in its IKEv2-SCSI header, then:

a) The command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST and the additional sense code set to SA CREATION PARAMETER VALUE INVALID; and
b) The device server shall abandon the IKEv2-SCSI CCS (see 5.13.5).

If the application client receives an invalid message id field in the parameter data for a SECURITY PROTOCOL IN command, the application client should notify the device server that the IKEv2-SCSI CCS is being abandoned as described in 5.13.5.

The IKE LENGTH field contains the total number of bytes in the parameter data, including the IKEv2-SCSI header and all the IKEv2-SCSI payloads.

NOTE x9 - The contents of the IKE LENGTH field differ from those found in most SCSI length fields, however, they are consistent with the IKEv2 usage (see RFC 4306).

Each IKEv2-SCSI payload (see 7.7.3.5) contains specific data related to the operation being performed. A specific combination of IKEv2-SCSI payloads is needed for each operation (e.g., Key Exchange) as summarized in 5.13.4.2. The Encryption payload (see 7.7.3.6.2) nests one set of IKEv2-SCSI payloads inside another.

{{SA CREATION PARAMETER VALUE INVALID (used multiple times in this subclause) is a new additional sense code.}}

#### 7.7.3.5 IKEv2-SCSI payloads

##### 7.7.3.5.1 IKEv2-SCSI payload format

Each IKEv2-SCSI payload (see table x9) is composed of a header and data that is specific to the payload type.

**Table x9 — IKEv2-SCSI payload format**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| IKEv2-SCSI payload header | | | | | | | | |
| 0 | NEXT PAYLOAD | | | | | | | |
| 1 | CRIT | Reserved | | | | | | |
| 2 | (MSB) | IKE PAYLOAD LENGTH (n+1) | | | | | | |
| 3 | | | | | | | | (LSB) |
| IKEv2-SCSI payload-specific data | | | | | | | | |
| 4 | | | | | | | | |
| n | | | | | | | | |

The NEXT PAYLOAD field identifies the IKEv2-SCSI payload that follows this IKEv2-SCSI payload using one of the code values shown in table x8 (see 7.7.3.4).

If a device server receives an IKEv2-SCSI payload that it does not recognize (e.g., an IKEv2-SCSI payload identified by a next payload value of 01h) with the critical (CRIT) bit set to one, then:

a) The command shall be terminated with a CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to SA CREATION PARAMETER VALUE INVALID; and
b) The device server shall abandon the IKEv2-SCSI CCS (see 5.13.5).

{{SA CREATION PARAMETER VALUE INVALID is a new additional sense code.}}

If an application client receives an IKEv2-SCSI payload that it does not recognize with the CRIT bit set to one, then the application client should notify the device server that it is abandoning the IKEv2-SCSI CCS as described in 5.13.5.

If an application client or device server receives an IKEv2-SCSI payload that it does not recognize with the CRIT bit set to zero, then it may use the NEXT PAYLOAD field and the IKE PAYLOAD LENGTH field to skip processing of the unrecognized IKEv2-SCSI payload and continue processing at the next IKEv2-SCSI payload.

The IKE PAYLOAD LENGTH field contains the total number of bytes in the payload, including the IKEv2-SCSI payload header. The value in the IKE PAYLOAD LENGTH field need not be a multiple of two or four (i.e., no byte alignment is maintained among IKEv2-SCSI payloads).

NOTE x10 - The contents of the IKE PAYLOAD LENGTH field differ from those found in most SCSI length fields, however, they are consistent with the IKEv2 usage (see RFC 4306).

The format and contents of the IKEv2-SCSI payload-specific data depends on the value in the NEXT PAYLOAD field of:

a) The IKEv2-SCSI header (see 7.7.3.4), if this is the first IKEv2-SCSI payload in the parameter data; or

b) The previous IKEv2-SCSI payload, in all other cases.

#### 7.7.3.5.2 No Next payload

A NEXT PAYLOAD field that is set to No Next payload (i.e., 00h) specifies that no more IKEv2-SCSI payloads follow the current payload. The IKEv2-SCSI No Next payload contains no bytes and has no format.

#### 7.7.3.5.3 Key Exchange payload

The Key Exchange payload (see table x10) transfers Diffie-Hellman secure key exchange data between an application client and a device server or vice versa.

**Table x10 — Key Exchange payload format**

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

The NEXT PAYLOAD field, CRIT bit, and IKE PAYLOAD LENGTH field are described in 7.7.3.5.1.

The CRIT bit is set to one in the Key Exchange payload.

{{The following paragraph is best guess effort to translate what r4 says about the group number field.}}

The DIFFIE-HELLMAN GROUP NUMBER field contains the least significant 16 bits from ALGORITHM IDENTIFIER field in the D-H IKEv2-SCSI algorithm descriptor (see 7.7.3.6.5) in the IKEv2-SCSI Cryptographic Algorithms payload (see 7.7.3.5.13).

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 (i.e., 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 KEY EXCHANGE DATA field if necessary.

Diffie-Hellman exponential reuse and reuse of analogous Diffie-Hellman public values for Diffie-Hellman mechanisms not based on exponentiation is permitted as specified in RFC 4306. The freshness and randomness of the random nonces are critical to the security of IKEv2-SCSI when Diffie-Hellman exponentials and public values are reused (see RFC 4306).

#### 7.7.3.5.4 Identification – Application Client payload and Identification – Device Server payload

The Identification – Application Client payload transfers identification information from the application client to the device server. The Identification – Device Server payload transfers identification information from the device server to the application client.

The Identification – Application Client payload and Identification – Device Server payload are formatted as follows:

1) The IKEv2-SCSI payload header (i.e., the NEXT PAYLOAD field, CRIT bit, and IKE PAYLOAD LENGTH field) is formatted and processed as described in 7.7.3.5.1;
2) The CRIT bit is set to one in the Identification payload; and
3) The IKEv2-SCSI payload-specific data is formatted and processed as is described for the Identification payload in RFC 4306, with the additional requirements described in this subclause.

The ID Type shall be one of the following:

a) If the parameter data also contains a Certificate payload (see ), the ID_DER_ASN1_DN and ID_DER_ASN1_GN may be used as follows:
   A) If the identity is the value of a certificate subject field (see RFC 3280), then ID_DER_ASN1_DN shall be used; or
   B) If the identity is the value of a name contained in a Subject Alternative Name (SubjectAltName) certificate extension (see RFC 3280), the ID_DER_ASN1_GN shall be used;

   or
b) If the parameter data does not contain a Certificate payload, one of the following ID Types should be used:
   A) ID_KEY_ID allows arbitrary identity data to be passed. SCSI port and device names may be passed using this type; or
   B) 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 the Certificate payload is included in the parameter data, the identity in the Identification – Application Client payload or Identification – Device Server payload is not required to match anything in the Certificate payload (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 payload.

If a device server receives an Identification – Application Client payload that does not conform to the requirements in RFC 4306 or the requirements in this subclause, then:

a) The command shall be terminated with a CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to SA CREATION PARAMETER VALUE INVALID; and
b) The device server shall abandon the IKEv2-SCSI CCS (see 5.13.5).

{{SA CREATION PARAMETER VALUE INVALID is a new additional sense code.}}

If an application client receives an Identification – Device Server payload that does not conform to the requirements in RFC 4306 or the requirements in this subclause, then the application client should notify the device server that it is abandoning the IKEv2-SCSI CCS as described in 5.13.5.

#### 7.7.3.5.5 Certificate payload and Certificate Request payload

The Certificate Request payload allows an application client or device server to request the use of certificates as part of identity authentication and to name one or more trust anchors for the certificate verification process. The

Certificate payload delivers a requested identity authentication certificate. The protocol for using Certificate Request payloads and Certificate payloads is described in 5.13.4.3.

The Certificate payload and Certificate Request payload are formatted as follows:

1) The IKEv2-SCSI payload header (i.e., the NEXT PAYLOAD field, CRIT bit, and IKE PAYLOAD LENGTH field) is formatted and processed as described in 7.7.3.5.1;
2) The CRIT bit is set to one in the Certificate payload and Certificate Request payload; and
3) The IKEv2-SCSI payload-specific data is formatted and processed as is described for the Identification payload in RFC 4306, with the additional requirements described in this subclause.

The Certificate Encodings shall be used as described in table x11.

**Table x11 — Certificate Encodings usage**

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

The relationship between the Certificate payload and the Identification payload is described in 7.7.3.5.4.

The CRIT bit is set to one in the Certificate payload and Certificate Request payload.

#### 7.7.3.5.6 Authentication payload

The Authentication payload (see table x12) allows the application client and a device server to verify that the data transfers in their IKEv2-SCSI CCS have not be compromised by a man-in-the-middle attack (see 5.13.1.4).

**Table x12 — Authentication payload format**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | NEXT PAYLOAD | | | | | | | |
| 1 | CRIT | Reserved | | | | | | |
| 2 | (MSB) | IKE PAYLOAD LENGTH (n+1) | | | | | | |
| 3 | | | | | | | | (LSB) |
| 4 | AUTH METHOD | | | | | | | |
| 5 | Reserved | | | | | | | |
| 7 | | | | | | | | |
| 8 | AUTHENTICATION DATA | | | | | | | |
| n | | | | | | | | |

The NEXT PAYLOAD field, CRIT bit, and IKE PAYLOAD LENGTH field are described in 7.7.3.5.1.

The CRIT bit is set to one in the Authentication payload.

The AUTH METHOD field contains the least significant 8 bits from ALGORITHM IDENTIFIER field in the SA_AUTH IKEv2-SCSI algorithm descriptor (see 7.7.3.6.6) in the IKEv2-SCSI Cryptographic Algorithms payload (see 7.7.3.5.13).

The value in the AUTH METHOD field specifies the algorithm to be used in computing the contents of the AUTHENTICATION DATA field as follows:

a) The algorithm is processed as described in RFC 4306 and the reference listed in table x13, but
b) The 17 ASCII character non-terminated pre-shared secret pad string "Key Pad for IKEv2" specified by RFC 4306 is replaced by the 22 ASCII character non-terminated pre-shared secret pad string "Key Pad for IKEv2-SCSI".

**Table x13 — AUTH METHOD references**

| Code | Description | Reference |
|---|---|---|
| 01h | RSA Digital Signature | RFC 4306 |
| 02h | Shared Key Message Integrity Code | RFC 4306 |
| 09h | ECDSA with SHA-256 on the P-256 curve | RFC 4754 |
| 0Ah | ECDSA with SHA-384 on the P-384 curve | RFC 4754 |
| 0Bh | ECDSA with SHA-512 on the P-521 curve | RFC 4754 |

In the Authentication step SECURITY PROTOCOL OUT command (see 5.13.4.8.2) parameter list, the AUTHENTICATION DATA field contains the result of applying the algorithm specified by the AUTH METHOD field as described in this subclause to the following concatenation of bytes:

1) All the bytes in the Data-In Buffer returned by the Device Server Capabilities step (see 5.13.4.6) SECURITY PROTOCOL IN command;
2) All the bytes in the Data-Out Buffer sent by the Key Exchange step SECURITY PROTOCOL OUT command (see 5.13.4.7.2) for which GOOD status was returned;
3) All the bytes in the Data-In Buffer returned by the most recent Key Exchange step SECURITY PROTOCOL IN command (see 5.13.4.7.3) for which GOOD status was returned;
4) All the bytes in the NONCE DATA field of the Nonce payload (see 7.7.3.5.7) that was received in the Key Exchange step SECURITY PROTOCOL IN command; and
5) All the bytes produced by applying the PRF selected by the PRF IKEv2-SCSI cryptographic algorithm descriptor (see 7.7.3.6.3) in the IKEv2-SCSI Cryptographic Algorithms payload (see 7.7.3.5.13) to the following inputs:
    1) The SK_pi secure key (see 5.13.4.4); and
    2) All the bytes in the IKEv2-SCSI payload-specific data part (see 7.7.3.5.1) of the Identification – Application Client payload (see 7.7.3.5.4).

When processing the Authentication step SECURITY PROTOCOL OUT command, the device server shall compute the expected contents of the AUTHENTICATION DATA field by applying the algorithm specified by the AUTH METHOD field as described in this subclause to the following concatenation of bytes:

1) All the bytes in the Data-In Buffer that the device server returned to any application client in response the most recently received the Device Server Capabilities step SECURITY PROTOCOL IN command;
2) All the bytes in the Data-Out Buffer received in the Key Exchange step SECURITY PROTOCOL OUT command (see 5.13.4.7.2) for which GOOD status was returned;
3) All the bytes in the Data-In Buffer sent by the most recent Key Exchange step SECURITY PROTOCOL IN command (see 5.13.4.7.3) for which GOOD status was returned;

4) All the bytes in the NONCE DATA field of the Nonce payload (see 7.7.3.5.7) that was sent in the Key Exchange step SECURITY PROTOCOL IN command; and
5) All the bytes produced by applying the PRF selected by the PRF IKEv2-SCSI cryptographic algorithm descriptor (see 7.7.3.6.3) in the IKEv2-SCSI Cryptographic Algorithms payload (see 7.7.3.5.13) to the following inputs:
   1) The SK_pi secure key (see 5.13.4.4); and
   2) All the bytes received in the IKEv2-SCSI payload-specific data part (see 7.7.3.5.1) of the Identification – Application Client payload (see 7.7.3.5.4) of the parameter list being processed.

If the expected contents of the AUTHENTICATION DATA field do not match actual contents of the AUTHENTICATION DATA field, then:

a) 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 UNABLE TO DECRYPT PARAMETER LIST; and
b) The device server shall continue the IKEv2-SCSI CCS by preparing to receive another Authentication step SECURITY PROTOCOL OUT command.

{{UNABLE TO DECRYPT PARAMETER LIST is a new additional sense code.}}

For the Authentication step SECURITY PROTOCOL IN command (see 5.13.4.8.3) parameter list, the device server shall compute the AUTHENTICATION DATA field contains by applying the algorithm specified by the AUTH METHOD field as described in this subclause to the following concatenation of bytes:

1) All the bytes in the Data-In Buffer that the device server returned to any application client in response the most recently received the Device Server Capabilities step SECURITY PROTOCOL IN command;
2) All the bytes in the Data-Out Buffer received in the Key Exchange step SECURITY PROTOCOL OUT command (see 5.13.4.7.2) for which GOOD status was returned;
3) All the bytes in the Data-In Buffer sent by the most recent Key Exchange step SECURITY PROTOCOL IN command (see 5.13.4.7.3) for which GOOD status was returned;
4) All the bytes in the NONCE DATA field of the Nonce payload (see 7.7.3.5.7) that was received in the Key Exchange step SECURITY PROTOCOL OUT command; and
5) All the bytes produced by applying the PRF selected by the PRF IKEv2-SCSI cryptographic algorithm descriptor (see 7.7.3.6.3) in the IKEv2-SCSI Cryptographic Algorithms payload (see 7.7.3.5.13) to the following inputs:
   1) The SK_pr secure key (see 5.13.4.4); and
   2) All the bytes in the IKEv2-SCSI payload-specific data part (see 7.7.3.5.1) of the Identification – Device Server payload (see 7.7.3.5.4).

After GOOD status is received for the Authentication step SECURITY PROTOCOL IN command, the application client should compute the expected contents of the AUTHENTICATION DATA field by applying the algorithm specified by the AUTH METHOD field as described in this subclause to the following concatenation of bytes:

1) All the bytes in the Data-In Buffer returned by the Device Server Capabilities step (see 5.13.4.6) SECURITY PROTOCOL IN command;
2) All the bytes in the Data-Out Buffer sent by the Key Exchange step SECURITY PROTOCOL OUT command (see 5.13.4.7.2) for which GOOD status was returned;
3) All the bytes in the Data-In Buffer returned by the most recent Key Exchange step SECURITY PROTOCOL IN command (see 5.13.4.7.3) for which GOOD status was returned;
4) All the bytes in the NONCE DATA field of the Nonce payload (see 7.7.3.5.7) that were sent in the Key Exchange step SECURITY PROTOCOL OUT command; and
5) All the bytes produced by applying the PRF selected by the PRF IKEv2-SCSI cryptographic algorithm descriptor (see 7.7.3.6.3) in the IKEv2-SCSI Cryptographic Algorithms payload (see 7.7.3.5.13) to the following inputs:

1) The SK_pr secure key (see 5.13.4.4); and
2) All the bytes in the IKEv2-SCSI payload-specific data part (see 7.7.3.5.1) of the Identification – Device Server payload (see 7.7.3.5.4) received in the SECURITY PROTOCOL IN parameter data.

f the expected contents of the AUTHENTICATION DATA field do not match actual contents of the AUTHENTICATION DATA field, then application client should notify the device server that it is abandoning the IKEv2-SCSI CCS as described in 5.13.5.

If the AUTH METHOD field contains 02h (i.e., Shared Key Message Integrity Code) the following requirements apply in addition to those found in RFC 4306:

a) A pre-shared secrets shall be associated with one identity;
b) The same pre-shared secret shall not be used to authenticate both an application client and a device server;
c) Use of the same pre-shared secret for a group of application clients or a group of device servers is strongly discouraged, because it enables any member of the group to impersonate any other member;
d) The means for provisioning pre-shared secrets are outside the scope of this standard;
e) The pre-shared secrets may be provisioned as follows:
   A) At the time of manufacturing;
   B) During device or system initialization; or
   C) Any time thereafter;
f) The following requirements from RFC 4306 apply to the interfaces for provisioning pre-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 it is used as a pre-shared secret;
   C) A hexadecimal ASCII encoding of the pre-shared secret shall be supported; and
   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;

   and
g) Information about the size of the pre-shared secret shall be stored at the same time that the pre-shared secret is stored.

If the AUTH METHOD field contains 01h (i.e., RSA Digital Signature) the following requirements apply in addition to those found in RFC 4306:

a) An RSA digital signature shall be encoded with the EMSA-PKCS1-v1_5 signature encoding method as specified in RFC 2437 (see RFC 4718); and
b) The RFC 4718 description of the bytes (i.e., octets) to be signed applies to IKEv2-SCSI with the following changes and clarifications:
   A) The InitiatorSignedOctets are signed by the application client;
   B) An IKEv2-SCSI SA Creation Capabilities payload (see 7.7.3.5.12) is prepended to the Responder-SignedOctets as described in this subclause and the result is signed by the device server;
   C) GenIKEHDR does not apply.
   D) The statement "[ four octets 0 if using port 4500 ]" does not exist in IKEv2-SCSI;
   E) SPIi is the application client SAI;
   F) SPIr is the device server SAI;
   G) RESERVED refers to a reserved field in the Identification payload (see RFC 4306).

#### 7.7.3.5.7 Nonce payload

The Nonce payload (see table x14) transfers one random nonce (see 3.1.95) from the application client to the device server and another from the device server to the application client.

**Table x14 — Nonce payload format**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | NEXT PAYLOAD | | | | | | | |
| 1 | CRIT | Reserved | | | | | | |
| 2 | (MSB) | IKE PAYLOAD LENGTH (n+1) | | | | | | |
| 3 | | | | | | | | (LSB) |
| 4 | NONCE DATA | | | | | | | |
| n | | | | | | | | |

The NEXT PAYLOAD field, CRIT bit, and IKE PAYLOAD LENGTH field are described in 7.7.3.5.1.

The CRIT bit is set to one in the Nonce payload.

In the Key Exchange step SECURITY PROTOCOL OUT command (see 5.13.4.7.2) the NONCE DATA field contains the application client's random nonce.

In the Key Exchange step SECURITY PROTOCOL IN command (see 5.13.4.7.3) the NONCE DATA field contains the device server's random nonce.

The requirements that RFC 4306 places on the nonce data shall apply to this standard.

#### 7.7.3.5.8 Notify payload

This standard uses the Notify payload (see table x15) only to provide initial contact notification from the application client to the device server.

**Table x15 — Notify payload format**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | NEXT PAYLOAD | | | | | | | |
| 1 | CRIT | Reserved | | | | | | |
| 2 | (MSB) | IKE PAYLOAD LENGTH (10h) | | | | | | |
| 3 | | | | | | | | (LSB) |
| 4 | PROTOCOL ID (01h) | | | | | | | |
| 5 | SAI SIZE (08h) | | | | | | | |
| 6 | (MSB) | NOTIFY MESSAGE TYPE (0662h) | | | | | | |
| 7 | | | | | | | | (LSB) |
| 8 | SAI | | | | | | | |
| 15 | | | | | | | | |

The NEXT PAYLOAD field, CRIT bit, and IKE PAYLOAD LENGTH field are described in 7.7.3.5.1.

The CRIT bit is set to one in the Notify payload.

The PROTOCOL ID field contains one. If the device server receives a protocol id other than one, then:

a) The command shall be terminated with a CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to SA CREATION PARAMETER VALUE INVALID; and
b) The device server shall abandon the IKEv2-SCSI CCS (see 5.13.5).

The SAI SIZE field contains eight. If the device server receives a value other than eight in the SAI SIZE field, then:

a) The command shall be terminated with a CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to SA CREATION PARAMETER VALUE INVALID; and
b) The device server shall abandon the IKEv2-SCSI CCS (see 5.13.5).

The NOTIFY MESSAGE TYPE field contains 16 384 (i.e., INITIAL_CONTACT). If the device server receives a value other than 16 384 in the NOTIFY MESSAGE TYPE field, then:

a) The command shall be terminated with a CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to SA CREATION PARAMETER VALUE INVALID; and
b) The device server shall abandon the IKEv2-SCSI CCS (see 5.13.5).

The SAI field contains the device server's SAI. If the contents of the SAI field are not identical to the contents of the IKE_SA DEVICE SERVER SAI field in the IKEv2-SCSI header (see 7.7.3.4), then:

a) The command shall be terminated with a CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to SA CREATION PARAMETER VALUE INVALID; and
b) The device server shall abandon the IKEv2-SCSI CCS (see 5.13.5).

Unless an error is detected, the device server shall process the Notify payload as described in 5.13.4.8.2.

{{SA CREATION PARAMETER VALUE INVALID (used multiple times in this subclause) is a new additional sense code.}}

#### 7.7.3.5.9 Delete payload

{{Several of the requirements in this subclause differ substantially with what r4 implies, but I think they are more sane.}}

The Delete payload requests the deletion of an existing SA or the abandonment of an IKEv2-SCSI CCS that is in progress. The device server shall ignore an Notify payload that is not received inside a valid Encrypted payload (see 7.7.3.5.11).

**Table x16 — Notify payload format**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | NEXT PAYLOAD | | | | | | | |
| 1 | CRIT | Reserved | | | | | | |
| 2 | (MSB) | IKE PAYLOAD LENGTH (18h) | | | | | | |
| 3 | | | | | | | | (LSB) |
| 4 | PROTOCOL ID (01h) | | | | | | | |
| 5 | SAI SIZE (08h) | | | | | | | |
| 6 | (MSB) | NUMBER OF SAIS (0002h) | | | | | | |
| 7 | | | | | | | | (LSB) |
| 8 | AC_SAI | | | | | | | |
| 15 | | | | | | | | |
| 16 | DC_SAI | | | | | | | |
| 23 | | | | | | | | |

The NEXT PAYLOAD field, CRIT bit, and IKE PAYLOAD LENGTH field are described in 7.7.3.5.1.

The CRIT bit is set to one in the Delete payload.

The PROTOCOL ID field contains one. If the device server receives a protocol id other than one, then:

a) The command shall be terminated with a CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to SA CREATION PARAMETER VALUE INVALID; and
b) If the contents of the APPLICATION CLIENT SAI field and DEVICE SERVER SAI field match the SAIs for an IKEv2-SCSI CCS for which the device server is maintaining state, the IKEv2-SCSI CCS shall be abandoned as described in 5.13.5.

The SAI SIZE field contains eight. If the device server receives a value other than eight in the SAI SIZE field, then:

a) The command shall be terminated with a CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to SA CREATION PARAMETER VALUE INVALID; and
b) If the contents of the APPLICATION CLIENT SAI field and DEVICE SERVER SAI field match the SAIs for an IKEv2-SCSI CCS for which the device server is maintaining state, the IKEv2-SCSI CCS shall be abandoned as described in 5.13.5.

The AC_SAI field contains the AC_SAI SA parameter value (see 3.1.103) for the SA to be deleted.

The DC_SAI field contains the DC_SAI SA parameter value (see 3.1.103) for the SA to be deleted.

If the device server is maintaining SA parameters for which the AC_SAI matches the contents of the AC_SAI field and the DC_SAI matches the contents of the DC_SAI field, that set of SA parameters shall be deleted.

If the device server is maintaining state for an IKEv2-SCSI CCS, the IKEv2-SCSI CCS shall be abandoned (see 5.13.5) if:

a) The contents of the AC_SAI field match the application client's SAI in the maintained state; and
b) The contents of the DC_SAI field match the device server's SAI in the maintained state.

{{Attempting to delete an SA based on only the AC_SAI (as r4 describes) would tie the deletion request to the same I_T_L nexus that created the SA. This is problematic and violates the currently published (SPC-4 r11) SA model. This revision requires both SAIs for deletion. This change conforms to the SA model and eliminates the I_T_L nexus linkage. The RFC 4306 description of the Delete payload is sufficiently vague to allow this usage. A note about the difference between this standard and RFC 4306 (see Annex C item q) has been added in this revision.}}

{{SA CREATION PARAMETER VALUE INVALID (used multiple times in this subclause) is a new additional sense code.}}

#### 7.7.3.5.10 Vendor ID payload

The Vendor ID payload allows the IKEv2-SCSI protocol to be extended in vendor specific way (see RFC 4306).

The Vendor ID is formatted as follows:

1) The IKEv2-SCSI payload header (i.e., the NEXT PAYLOAD field, CRIT bit, and IKE PAYLOAD LENGTH field) is formatted and processed as described in 7.7.3.5.1;
2) The CRIT bit is set to zero in the Vendor ID payload; and
3) The IKEv2-SCSI payload-specific data is formatted and processed as described is described for the Vendor ID payload in RFC 4306, with the additional requirements described in this subclause.

The RFC 4306 paragraph on the topic of Internet-Drafts does not apply to IKEv2-SCSI.

#### 7.7.3.5.11 Encrypted payload

##### 7.7.3.5.11.1 Introduction

The Encrypted payload transfers one or more other IKEv2-SCSI payloads that are encrypted and integrity checked from the application client to the device server and vice versa.

The Encrypted payload is formatted as follows:

1) The IKEv2-SCSI payload header (i.e., the NEXT PAYLOAD field, CRIT bit, and IKE PAYLOAD LENGTH field) is formatted and processed as described in 7.7.3.5.1;
2) The NEXT PAYLOAD field in the Encrypted payload specifies the type of the first IKEv2-SCSI payload in the encrypted data;
3) The CRIT bit is set to one in the Encrypted payload; and
4) The IKEv2-SCSI payload-specific data is formatted and processed as described is described for the Encrypted payload in RFC 4306, with the additional requirements described in this subclause.

Before performing any checks of data contained in the Encrypted payload, the contents of the Encrypted payload are decrypted and integrity checked based on the contents of the IKE_SA APPLICATION CLIENT SAI field and the IKE_SA DEVICE SERVER SAI field in the IKEv2-SCSI header (see 7.7.3.4) as follows:

a) If a device server receives an Encrypted payload in a SECURITY PROTOCOL OUT command parameter list, then:
   A) If the contents of the IKE_SA APPLICATION CLIENT SAI field match the AC_SAI SA parameter (see 3.1.103) in a generated SA and the IKE_SA DEVICE SERVER SAI field match the DC_SAI SA parameter in the same generated SA, then the contents of the MGMT_DATA SA parameter are used to decrypt and integrity check the Encrypted payload as described in ;
   B) If the device server is maintaining state for an IKEv2-CCS the payload is decrypted as described in 7.7.3.5.11.3 if:
      a) The contents of the IKE_SA APPLICATION CLIENT SAI field match the application client's SAI in the maintained IKEv2-SCSI CCS state; and
      b) The contents of the IKE_SA DEVICE SERVER SAI field match the device server's SAI in the maintained IKEv2-SCSI CCS state;

      or
   C) In all other cases, the command is processed as follows:
      a) 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 UNABLE TO DECRYPT PARAMETER LIST; and
      b) If state is being maintained for an IKEv2-SCSI CCS, the device server shall continue the IKEv2-SCSI CCS by preparing to receive another Authentication step SECURITY PROTOCOL OUT command.

   and
b) If an application client receives an Encrypted payload in a SECURITY PROTOCOL IN command parameter list, then the payload is decrypted and integrity checked as described in 7.7.3.5.11.4.

{{UNABLE TO DECRYPT PARAMETER LIST is a new additional sense code.}}

**7.7.3.5.11.2 Decrypting an Encrypted payload that is not part of an IKEv2-SCSI CCS**

Before performing any checks of data contained in the Encrypted payload received in a SECURITY PROTCOL OUT command parameter list (see 5.13.4.8.2) for a generated SA, the device server shall decrypt and check the integrity of the Encrypted payload as follows:

1) Decrypt the encrypted portion of the Encrypted payload using the encryption algorithm, key length, and SK_ei secure key (see 5.13.4.4) stored in the MGMT_DATA SA parameter (see 5.13.4.9); and
2) Integrity check the decrypted data as follows:
   1) Use the algorithm specified by the integrity algorithm and SK_ai secure key (see 5.13.4.4) stored in the MGMT_DATA SA parameter to compute an integrity check value for the decrypted data; and
   2) Verify that the computed integrity check value matches the one found in the Encrypted payload.

If the integrity checking of the Encrypted payload fails, then 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 UNABLE TO DECRYPT PARAMETER LIST.

{{UNABLE TO DECRYPT PARAMETER LIST is a new additional sense code.}}

#### 7.7.3.5.11.3 Decrypting an Authentication step SECURITY PROTOCOL OUT command Encrypted payload

Before performing any checks of data contained in the Encrypted payload received in a in the Authentication step SECURITY PROTOCOL OUT parameter list (see 5.13.4.8.2), the device server shall decrypt and check the integrity of the Encrypted payload as follows:

1) Decrypt the encrypted portion of the Encrypted payload using the algorithm and key length specified by the ENCR IKEv2-SCSI algorithm descriptor (see 7.7.3.6.2) in the IKEv2-SCSI Cryptographic Algorithms payload (see 7.7.3.5.13) and the SK_ei secure key (see 5.13.4.4); and
2) Integrity check the decrypted data as follows:
    1) Use the algorithm specified by the INTEG IKEv2-SCSI algorithm descriptor (see 7.7.3.6.4) in the IKEv2-SCSI Cryptographic Algorithms payload and the SK_ai secure key (see 5.13.4.4) to compute an integrity check value for the decrypted data; and
    2) Verify that the computed integrity check value matches the one found in the Encrypted payload.

If the integrity checking of the Encrypted payload fails, then:

a) 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 UNABLE TO DECRYPT PARAMETER LIST; and
b) The device server shall continue the IKEv2-SCSI CCS by preparing to receive another Authentication step SECURITY PROTOCOL OUT command.

{{UNABLE TO DECRYPT PARAMETER LIST is a new additional sense code.}}

#### 7.7.3.5.11.4 Decrypting an Authentication step SECURITY PROTOCOL IN command Encrypted payload

Before performing any checks of data contained in the Encrypted payload received in a in the Authentication step SECURITY PROTOCOL IN parameter list (see 5.13.4.8.3), the application client should decrypt and check the integrity of the Encrypted payload as follows:

1) Decrypt the encrypted portion of the Encrypted payload using the algorithm specified by the ENCR IKEv2-SCSI algorithm descriptor (see 7.7.3.6.2) in the IKEv2-SCSI Cryptographic Algorithms payload (see 7.7.3.5.13) and the SK_er secure key (see 5.13.4.4); and
2) Integrity check the decrypted data as follows:
    1) Use the algorithm specified by the INTEG IKEv2-SCSI algorithm descriptor (see 7.7.3.6.4) in the IKEv2-SCSI Cryptographic Algorithms payload and the SK_ar secure key (see 5.13.4.4) to compute an integrity check value for the decrypted data; and
    2) Verify that the computed integrity check value matches the one found in the Encrypted payload.

If the integrity checking of the Encrypted payload fails, then the application client should notify the device server that it is abandoning the IKEv2-SCSI CCS as described in 5.13.5.

#### 7.7.3.5.12 IKEv2-SCSI SA Creation Capabilities payload

The IKEv2-SCSI SA Creation Capabilities payload (see table x17) lists all the security algorithms that the device server allows to be used in an IKEv2-SCSI CCS. Events that are outside the scope of this standard may change the contents of the IKEv2-SCSI SA Creation Capabilities payload at any time.

**Table x17 — IKEv2-SCSI SA Creation Capabilities payload format**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | NEXT PAYLOAD | | | | | | | |
| 1 | CRIT | Reserved | | | | | | |
| 2 | (MSB) | IKE PAYLOAD LENGTH (n+1) | | | | | | |
| 3 | | | | | | | | (LSB) |
| 4 | Reserved | | | | | | | |
| 6 | | | | | | | | |
| 7 | NUMBER OF ALGORITHM DESCRIPTORS | | | | | | | |
| IKEv2-SCSI cryptographic algorithm descriptors | | | | | | | | |
| 8 | First IKEv2-SCSI cryptographic algorithm descriptor (see 7.7.3.6) | | | | | | | |
| | | | | | | | | |
| | . . . | | | | | | | |
| | Last IKEv2-SCSI cryptographic algorithm descriptor (see 7.7.3.6) | | | | | | | |
| n | | | | | | | | |

The NEXT PAYLOAD field, CRIT bit, and IKE PAYLOAD LENGTH field are described in 7.7.3.5.1.

The CRIT bit is set to one in the IKEv2-SCSI SA Creation Capabilities payload.

The NUMBER OF ALGORITHM DESCRIPTORS field contains the number of IKEv2-SCSI cryptographic algorithm descriptors that follow in the IKEv2-SCSI SA Creation Capabilities payload.

Each IKEv2-SCSI cryptographic algorithm descriptor (see 7.7.3.6) describes one combination of security algorithm and algorithm attributes that the device server allows to be used in an IKEv2-SCSI CCS. If more than one set of algorithm attributes (e.g., key length) is allowed for any allowed security algorithm, a different SCSI cryptographic algorithms descriptor shall be included for each set of algorithm attributes.

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, if any, within the same algorithm identifier.

The algorithms allowed may be a subset of the algorithms supported by the device server.

The method for changing which of the device server supported algorithms are allowed is outside the scope of this standard, but changes in allowed algorithms do not take effect until the new list is returned to an application client (any application client) in an IKEv2-SCSI SA Creation Capabilities payload.

#### 7.7.3.5.13 IKEv2-SCSI Cryptographic Algorithms payload

The IKEv2-SCSI Cryptographic Algorithms payload (see table x18) lists the security algorithms that are being used in an IKEv2-SCSI CCS.

**Table x18 — IKEv2-SCSI Cryptographic Algorithms payload format**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | NEXT PAYLOAD | | | | | | | |
| 1 | CRIT | Reserved | | | | | | |
| 2 | (MSB) | IKE PAYLOAD LENGTH (n+1) | | | | | | |
| 3 | | | | | | | | (LSB) |
| 4 | (MSB) | SA TYPE | | | | | | |
| 5 | | | | | | | | (LSB) |
| 6 | (MSB) | USAGE DATA LENGTH (j) | | | | | | |
| 7 | | | | | | | | (LSB) |
| 8 | (MSB) | SAI | | | | | | |
| 15 | | | | | | | | (LSB) |
| 16 | | USAGE DATA | | | | | | |
| 16+j-1 | | | | | | | | |
| 16+j | | Reserved | | | | | | |
| 16+j+2 | | | | | | | | |
| 16+j+3 | NUMBER OF ALGORITHM DESCRIPTORS | | | | | | | |
| IKEv2-SCSI cryptographic algorithm descriptors | | | | | | | | |
| 16+j+4 | First IKEv2-SCSI cryptographic algorithm descriptor (see 7.7.3.6) | | | | | | | |
| | . . . | | | | | | | |
| n | Last IKEv2-SCSI cryptographic algorithm descriptor (see 7.7.3.6) | | | | | | | |

The NEXT PAYLOAD field, CRIT bit, and IKE PAYLOAD LENGTH field are described in 7.7.3.5.1.

The CRIT bit is set to one in the IKEv2-SCSI Cryptographic Algorithms payload.

The SA TYPE field specifies the usage type for the SA and is selected from among those listed in table 45 (see 5.13.2.2). If a device server receives an SA TYPE field that contains an SA usage type whose use the device server does not allow, then:

a) The command shall be terminated with a CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to SA CREATION PARAMETER VALUE INVALID; and
b) The device server shall abandon the IKEv2-SCSI CCS (see 5.13.5).

The method for changing which of the device server supported SA usage types are allowed is outside the scope of this standard.

The USAGE DATA LENGTH field specifies number of bytes of usage data that follow the eight-byte said field.

NOTE x11 - The contents of the USAGE DATA LENGTH field differ from those found in most SCSI length fields, however, they are consistent with the IKEv2 usage (see RFC 4306).

The SAID field is reserved.

The USAGE DATA field contains information that will be stored in the USAGE_DATA SA parameter (see 3.1.103) if the SA is generated (see 5.13.4.9).

The NUMBER OF ALGORITHM DESCRIPTORS field contains the number of IKEv2-SCSI cryptographic algorithm descriptors that follow in the IKEv2-SCSI Cryptographic Algorithms payload. If a device server receives NUMBER OF ALGORITHM DESCRIPTORS field that contains a value other than five, then:

a) The command shall be terminated with a CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to SA CREATION PARAMETER VALUE INVALID; and
b) The device server shall abandon the IKEv2-SCSI CCS (see 5.13.5).

Each IKEv2-SCSI cryptographic algorithm descriptor (see 7.7.3.6) describes one combination of security algorithm and algorithm attributes during the IKEv2-SCSI CCS. The IKEv2-SCSI cryptographic algorithm descriptors are ordered as follows:

1) One ENCR IKEv2-SCSI cryptographic algorithm descriptor (see 7.7.3.6.2);
2) One PRF IKEv2-SCSI cryptographic algorithm descriptor (see 7.7.3.6.3);
3) One INTEG IKEv2-SCSI cryptographic algorithm descriptor (see 7.7.3.6.4);
4) One D-H IKEv2-SCSI cryptographic algorithm descriptor (see 7.7.3.6.5); and
5) One SA_AUTH IKEv2-SCSI cryptographic algorithm descriptor (see 7.7.3.6.6).

If a device server receives an IKEv2-SCSI Cryptographic Algorithms payload that does not contain the IKEv2-SCSI cryptographic algorithm descriptors described in this subclause in the order described in this subclause, then:

a) The command shall be terminated with a CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to SA CREATION PARAMETER VALUE INVALID; and
b) The device server shall abandon the IKEv2-SCSI CCS (see 5.13.5).

If a specific field is the cause for returning the SA CREATION PARAMETER VALUE INVALID additional sense code, then the SKSV bit shall be set to one and SENSE KEY SPECIFIC field shall be set as defined in 4.5.2.4.2.

In the Key Exchange step SECURITY PROTOCOL IN parameter data (see 5.13.4.7.3), the device server returns the IKEv2-SCSI Cryptographic Algorithms payload received during the Key Exchange step SECURITY PROTOCOL OUT command (see 5.13.4.7.2) to confirm acceptance of the algorithms.

{{SA CREATION PARAMETER VALUE INVALID (used multiple times in this subclause) is a new additional sense code.}}

#### 7.7.3.5.14 IKEv2-SCSI Timeout Values payload

The IKEv2-SCSI Timeout Values payload (see table x19) specifies the timeout intervals associated with an IKEv2-SCSI CCS

**Table x19 — IKEv2-SCSI Timeout Values payload format**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | NEXT PAYLOAD | | | | | | | |
| 1 | CRIT | Reserved | | | | | | |
| 2 | (MSB) | IKE PAYLOAD LENGTH (10h) | | | | | | |
| 3 | | | | | | | | (LSB) |
| 4 | Reserved | | | | | | | |
| 6 | | | | | | | | |
| 7 | NUMBER OF TIMEOUT VALUES (02h) | | | | | | | |
| 8 | (MSB) | IKEV2-SCSI PROTOCOL TIMEOUT | | | | | | |
| 11 | | | | | | | | (LSB) |
| 12 | (MSB) | IKEV2-SCSI SA INACTIVITY TIMEOUT | | | | | | |
| 15 | | | | | | | | (LSB) |

The NEXT PAYLOAD field, CRIT bit, and IKE PAYLOAD LENGTH field are described in 7.7.3.5.1.

The CRIT bit is set to one in the IKEv2-SCSI Timeout Values payload.

The NUMBER OF TIMEOUT VALUES field specifies the number of four-byte timeout values that follow. If the number of timeout values is less than the number the device server expects, the device server shall substitute a value of ten (i.e., ten seconds) for each missing timeout value.

The IKEV2-SCSI PROTOCOL TIMEOUT field specifies the number of seconds that the device server shall wait for the next command in the IKEv2-SCSI CCS. If the timeout expires before the device server receives an IKEv2-SCSI CCS command, the device server shall abandon the IKEv2-SCSI CCS as described in 5.13.5.

The IKEV2-SCSI SA INACTIVITY TIMEOUT field 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 SA parameter when the SA is generated (see 5.13.4.9).

The device server shall replace any timeout value that is set to zero with a value of the (i.e., ten seconds).

NOTE x12 - 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.

### 7.7.3.6 IKEv2-SCSI cryptographic algorithm descriptors

#### 7.7.3.6.1 Overview

Each IKEv2-SCSI cryptographic algorithm descriptor (see table x20) specifies one algorithm used for encryption, integrity checking, key generation, or authentication.

**Table x20 — IKEv2-SCSI cryptographic algorithm descriptor format**

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

The ALGORITHM TYPE field (see table x21) specifies the type of cryptographic algorithm to which the IKEv2-SCSI cryptographic algorithm descriptor applies.

**Table x21 — ALGORITHM TYPE field**

| Code | Name | Description | Reference |
|---|---|---|---|
| 01h | ENCR | Encryption algorithm | 7.7.3.6.2 |
| 02h | PRF | Pseudo-random function | 7.7.3.6.3 |
| 03h | INTEG | Integrity algorithm | 7.7.3.6.4 |
| 04h | D-H | Diffie-Hellman group | 7.7.3.6.5 |
| 05h - F0h | Restricted | | RFC 4306 |
| F9h | SA_AUTH | IKEv2-SCSI authentication algorithm | 7.7.3.6.6 |
| All others | Reserved | | |

The IKE DESCRIPTOR LENGTH field contains 12 (i.e., the total number of bytes in the IKEv2-SCSI cryptographic algorithms descriptor including the ALGORITHM TYPE field and reserved byte).

NOTE x13 - The contents of the IKE DESCRIPTOR LENGTH field differ from those found in most SCSI length fields, however, they are consistent with the IKEv2 usage (see RFC 4306).

The contents of the ALGORITHM IDENTIFIER field and ALGORITHM ATTRIBUTES field depend on the contents of the ALGORITHM TYPE field (see table x21). The algorithm attributes field is reserved in some IKEv2-SCSI cryptographic algorithms descriptor formats.

#### 7.7.3.6.2 Encryption algorithm (ENCR) IKEv2-SCSI cryptographic algorithm descriptors

When the algorithm type field is set to ENCR (i.e., 01h) in an IKEv2-SCSI cryptographic algorithm descriptor (see table x22), the descriptor specifies an encryption algorithm to be applied during the IKEv2-SCSI Authentication step (see 5.13.4.8) and when the SA created by the IKEv2-SCSI is applied to user data.

**Table x22 — ENCR IKEv2-SCSI cryptographic algorithm descriptor format**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | ALGORITHM TYPE (01h) | | | | | | | |
| 1 | Reserved | | | | | | | |
| 2 | (MSB) | | | IKE DESCRIPTOR LENGTH (000Ch) | | | | |
| 3 | | | | | | | | (LSB) |
| 4 | (MSB) | | | ALGORITHM IDENTIFIER | | | | |
| 7 | | | | | | | | (LSB) |
| 8 | | | | Reserved | | | | |
| 9 | | | | | | | | |
| 10 | (MSB) | | | KEY LENGTH | | | | |
| 11 | | | | | | | | (LSB) |

The ALGORITHM TYPE field and IKE DESCRIPTOR LENGTH field are described in 7.7.3.6.1.

The ALGORITHM IDENTIFIER field (see table x23) specifies the encryption algorithm to which the ENCR IKEv2-SCSI cryptographic algorithm descriptor applies.

**Table x23 — ENCR ALGORITHM IDENTIFIER field**

| Code | Description | Key length (bytes) | Support | Reference |
|---|---|---|---|---|
| 0001 000Bh | ENCR_NULL | 0 | Mandatory | |
| 0001 000Ch | AES-CBC | 16 | Mandatory | RFC 3602 |
| | | 24 | Prohibited | |
| | | 32 | Optional | |
| 0001 0010h | AES-CCM with a 16 byte MAC [a] | 16 | Optional | NIST SP 800-38C |
| | | 24 | Prohibited | |
| | | 32 | Optional | |

[a] AES-CCM is a combined mode algorithm that provides both encryption and cryptographic integrity. The AUTH_COMBINED integrity checking algorithm (see 7.7.3.6.4) is used with this algorithm. AES-CCM requires 3 bytes of keying material in addition to the AES key for use as a salt, see NIST SP 800-38C.

[b] AES-GCM is a combined mode algorithm that provides both encryption and cryptographic integrity. The AUTH_COMBINED integrity checking algorithm is used with this algorithm. AES-GCM requires 4 bytes of keying material in addition to the AES key for use as a salt, see NIST SP 800-38D.

**Table x23 — ENCR ALGORITHM IDENTIFIER field**

| Code | Description | Key length (bytes) | Support | Reference |
|---|---|---|---|---|
| 0001 0014h | AES-GCM with a 16 byte MAC [b] | 16 | Optional | NIST SP 800-38D |
| | | 24 | Prohibited | |
| | | 32 | Optional | |
| 0001 0400h – 0001 FFFFh | Vendor Specific | | | |
| 0000 0000h – 0000 FFFFh | Restricted | | | IANA |
| All others | Reserved | | | |

[a] AES-CCM is a combined mode algorithm that provides both encryption and cryptographic integrity. The AUTH_COMBINED integrity checking algorithm (see 7.7.3.6.4) is used with this algorithm. AES-CCM requires 3 bytes of keying material in addition to the AES key for use as a salt, see NIST SP 800-38C.

[b] AES-GCM is a combined mode algorithm that provides both encryption and cryptographic integrity. The AUTH_COMBINED integrity checking algorithm is used with this algorithm. AES-GCM requires 4 bytes of keying material in addition to the AES key for use as a salt, see NIST SP 800-38D.

{{Where do the additional bytes of salt described in the table x23 footnotes come from? What needs to be added to 5.13.4.7.4 to cover this need?}}

{{The next sentence is intended to rocket David Black into orbit … unless it happens to be true. If it is not true then, Ralph fails to grasp how encryption is used in the CCS but not used by the resulting SA.}}

ENCR_NULL indicates that encryption is not to be applied:

a) During the IKEv2-SCSI Authentication step (see 5.13.4.8.2); and
b) When the SA created by the IKEv2-SCSI is applied to user data.

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

A Key Exchange step SECURITY PROTOCOL OUT command (see 5.13.4.7.2) shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, the additional sense code set to SA CREATION PARAMETER VALUE INVALID the SKSV bit set to one, and SENSE KEY SPECIFIC field set as defined in 4.5.2.4.2, if the parameter list contains an IKEv2-SCSI Cryptographic Algorithms payload (see 7.7.3.5.13) that contains:

a) The following combination of IKEv2-SCSI cryptographic algorithm descriptors (see 7.7.3.6.4):
   A) An INTEG IKEv2-SCSI cryptographic algorithm descriptor with the algorithm identifier set to a value other than AUTH_COMBINED; and
   B) An ENCR IKEv2-SCSI cryptographic algorithm descriptor with the algorithm identifier set to a value that table x23 describes as requiring AUTH_COMBINED as the integrity check algorithm;

   or
b) The following combination of IKEv2-SCSI cryptographic algorithm descriptors:
   A) An INTEG IKEv2-SCSI cryptographic algorithm descriptor with the algorithm identifier set to AUTH_COMBINED; and
   B) An ENCR IKEv2-SCSI cryptographic algorithm descriptor with the algorithm identifier set to a value that table x23 does not describe as requiring AUTH_COMBINED as the integrity check algorithm.

The SA CREATION PARAMETER VALUE INVALID additional sense code indicates that the device server has abandoned the IKEv2-SCSI CCS (see 5.13.5).

The KEY LENGTH field specifies the number of bytes in the secure key (see 3.1.w) for the encryption algorithm to which the ENCR IKEv2-SCSI cryptographic algorithm descriptor applies. The same algorithm identifier may appear in multiple ENCR IKEv2-SCSI cryptographic algorithm descriptors with differing values in the key length field distinguishing the specified encryption algorithms.

A Key Exchange step SECURITY PROTOCOL OUT command (see 5.13.4.7.2) shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to SA CREATION PARAMETER VALUE INVALID, if the parameter list contains:

a) No ENCR IKEv2-SCSI cryptographic algorithm descriptors;
b) More than one ENCR IKEv2-SCSI cryptographic algorithm descriptor;
c) An ENCR IKEv2-SCSI cryptographic algorithm descriptor that does not appear in the SA Creations Capabilities payload (see 7.7.3.5.12) most recently returned by the device server to any application client; or
d) An ENCR IKEv2-SCSI cryptographic algorithm descriptor that contains:
   A) An algorithm identifier that is not shown in table x23; or
   B) A key length that:
      a) Does not match one of the values shown in table x23; or
      b) Is not supported by the device server.

If a specific field is the cause for returning the SA CREATION PARAMETER VALUE INVALID additional sense code, then the SKSV bit shall be set to one and SENSE KEY SPECIFIC field shall be set as defined in 4.5.2.4.2.

The SA CREATION PARAMETER VALUE INVALID additional sense code indicates that the device server has abandoned the IKEv2-SCSI CCS (see 5.13.5).

{{SA CREATION PARAMETER VALUE INVALID (used multiple times in this subclause) is a new additional sense code.}}

#### 7.7.3.6.3 Pseudo-random function (PRF) IKEv2-SCSI cryptographic algorithm descriptors

When the algorithm type field is set to PRF (i.e., 02h) in an IKEv2-SCSI cryptographic algorithm descriptor (see table x24), the descriptor specifies the pseudo-random function and KDF to be used during the Key Exchange step completion (see 5.13.4.7.4).

**Table x24 — PRF IKEv2-SCSI cryptographic algorithm descriptor format**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | ALGORITHM TYPE (02h) | | | | | | | |
| 1 | Reserved | | | | | | | |
| 2 | (MSB) | | | IKE DESCRIPTOR LENGTH (000Ch) | | | | |
| 3 | | | | | | | | (LSB) |
| 4 | (MSB) | | | ALGORITHM IDENTIFIER | | | | |
| 7 | | | | | | | | (LSB) |
| 8 | | | | Reserved | | | | |
| 11 | | | | | | | | |

The ALGORITHM TYPE field and IKE DESCRIPTOR LENGTH field are described in 7.7.3.6.1.

The ALGORITHM IDENTIFIER field (see table x25) specifies PRF and KDF to which the PRF IKEv2-SCSI cryptographic algorithm descriptor applies.

**Table x25 — PRF ALGORITHM IDENTIFIER field**

| Code | Description | Support | Reference | |
|---|---|---|---|---|
| | | | PRF | KDF |
| 0002 0002h | IKEv2-use based on SHA-1 | Optional | RFC 2104 | 5.13.3.3 |
| 0002 0004h | IKEv2-use based on AES-128 in CBC mode | Mandatory | RFC 4434 | 5.13.3.4 |
| 0002 0005h | IKEv2-use based on SHA-256 | Optional | RFC 4868 | 5.13.3.3 |
| 0002 0006h | IKEv2-use based on SHA-384 | Optional | RFC 4868 | 5.13.3.3 |
| 0002 0007h | IKEv2-use based on SHA-512 | Optional | RFC 4868 | 5.13.3.3 |
| 0002 0400h – 0002 FFFFh | Vendor Specific | | | |
| 0000 0000h – 0000 FFFFh | Restricted | | IANA | |
| All others | Reserved | | | |

A Key Exchange step SECURITY PROTOCOL OUT command (see 5.13.4.7.2) shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to SA CREATION PARAMETER VALUE INVALID, if the parameter list contains:

a) No PRF IKEv2-SCSI cryptographic algorithm descriptors;
b) More than one PRF IKEv2-SCSI cryptographic algorithm descriptor; or
c) A PRF IKEv2-SCSI cryptographic algorithm descriptor that does not appear in the SA Creations Capabilities payload (see 7.7.3.5.12) most recently returned by the device server to any application client; or
d) An PRF IKEv2-SCSI cryptographic algorithm descriptor that contains an algorithm identifier that is not shown in table x25.

If a specific field is the cause for returning the SA CREATION PARAMETER VALUE INVALID additional sense code, then the SKSV bit shall be set to one and SENSE KEY SPECIFIC field shall be set as defined in 4.5.2.4.2.

The SA CREATION PARAMETER VALUE INVALID additional sense code indicates that the device server has abandoned the IKEv2-SCSI CCS (see 5.13.5).

{{SA CREATION PARAMETER VALUE INVALID is a new additional sense code.}}

#### 7.7.3.6.4 Integrity algorithm (INTEG) IKEv2-SCSI cryptographic algorithm descriptors

When the algorithm type field is set to INTEG (i.e., 03h) in an IKEv2-SCSI cryptographic algorithm descriptor (see table x26), the descriptor specifies an integrity checking (i.e., data authentication) algorithm to be applied during the IKEv2-SCSI Authentication step (see 5.13.4.8) and when the SA created by the IKEv2-SCSI is applied to user data.

**Table x26 — INTEG IKEv2-SCSI cryptographic algorithm descriptor format**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | ALGORITHM TYPE (03h) | | | | | | | |
| 1 | Reserved | | | | | | | |
| 2 | (MSB) | | | IKE DESCRIPTOR LENGTH (000Ch) | | | | |
| 3 | | | | | | | | (LSB) |
| 4 | (MSB) | | | ALGORITHM IDENTIFIER | | | | |
| 7 | | | | | | | | (LSB) |
| 8 | | | | Reserved | | | | |
| 11 | | | | | | | | |

The ALGORITHM TYPE field and IKE DESCRIPTOR LENGTH field are described in 7.7.3.6.1.

The ALGORITHM IDENTIFIER field (see table x27) specifies integrity checking algorithm and secure key length to which the INTEG IKEv2-SCSI cryptographic algorithm descriptor applies.

**Table x27 — INTEG ALGORITHM IDENTIFIER field**

| Code | IKEv2 Name | Key length (bytes) | Support | Reference |
|---|---|---|---|---|
| 0003 0000h | AUTH_NONE | 0 | Optional | RFC 4306 |
| 0003 0002h | AUTH_HMAC_SHA1_96 | 20 | Optional | RFC 2404 |
| 0003 000Ch | AUTH_HMAC_SHA2_256_128 | 32 | Mandatory | RFC 4868 |
| 0003 000Dh | AUTH_HMAC_SHA2_384_192 | 48 | Optional | RFC 4868 |
| 0003 000Eh | AUTH_HMAC_SHA2_512_256 | 64 | Optional | RFC 4868 |
| F003 0001h | AUTH_COMBINED | 0 | Optional | this subclause |
| 0003 0400h – 0003 FFFFh | Vendor Specific | | | |
| 0000 0000h – 0000 FFFFh | Restricted | | | IANA |
| All others | Reserved | | | |

The AUTH_COMBINED integrity checking algorithm is used with encryption algorithms that include integrity checking as described in 7.7.3.6.2. In terms of an integrity check algorithm that is processed apart from the encryption algorithm, AUTH_COMBINED is equivalent to AUTH_NONE.

The key length used with an integrity checking algorithm is determined by the algorithm identifier as shown in table x27.

A Key Exchange step SECURITY PROTOCOL OUT command (see 5.13.4.7.2) shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to SA CREATION PARAMETER VALUE INVALID, if the parameter list contains:

a) No INTEG IKEv2-SCSI cryptographic algorithm descriptors;
b) More than one INTEG IKEv2-SCSI cryptographic algorithm descriptor; or
c) An INTEG IKEv2-SCSI cryptographic algorithm descriptor that does not appear in the SA Creations Capabilities payload (see 7.7.3.5.12) most recently returned by the device server to any application client; or
d) An INTEG IKEv2-SCSI cryptographic algorithm descriptor that contains an algorithm identifier that is not shown in table x27.

If a specific field is the cause for returning the SA CREATION PARAMETER VALUE INVALID additional sense code, then the SKSV bit shall be set to one and SENSE KEY SPECIFIC field shall be set as defined in 4.5.2.4.2.

The SA CREATION PARAMETER VALUE INVALID additional sense code indicates that the device server has abandoned the IKEv2-SCSI CCS (see 5.13.5).

{{SA CREATION PARAMETER VALUE INVALID is a new additional sense code.}}

### 7.7.3.6.5 Diffie-Hellman group (D-H) IKEv2-SCSI cryptographic algorithm descriptors

When the algorithm type field is set to D-H (i.e., 04h) in an IKEv2-SCSI cryptographic algorithm descriptor (see table x28), the descriptor specifies Diffie-Hellman group and Diffie-Hellman algorithm used during the IKEv2-SCSI Key Exchange step (see 5.13.4.7) to derive a secure key (see 3.1.w) that is known only to the application client and device server.

**Table x28 — D-H IKEv2-SCSI cryptographic algorithm descriptor format**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | ALGORITHM TYPE (04h) | | | | | | | |
| 1 | Reserved | | | | | | | |
| 2 | (MSB) | | | IKE DESCRIPTOR LENGTH (000Ch) | | | | |
| 3 | | | | | | | | (LSB) |
| 4 | (MSB) | | | ALGORITHM IDENTIFIER | | | | |
| 7 | | | | | | | | (LSB) |
| 8 | | | | Reserved | | | | |
| 11 | | | | | | | | |

The ALGORITHM TYPE field and IKE DESCRIPTOR LENGTH field are described in 7.7.3.6.1.

The ALGORITHM IDENTIFIER field (see table x29) specifies Diffie-Hellman algorithm, group, and secure key length to which the D-H IKEv2-SCSI cryptographic algorithm descriptor applies.

**Table x29 — D-H ALGORITHM IDENTIFIER field**

| Code | Description | Key length (bytes) | Support | Reference |
|---|---|---|---|---|
| 0004 000Eh | 2 048-bit MODP group (finite field D-H) | 256 | Mandatory | RFC 3526 |
| 0004 000Fh | 3 072-bit MODP group (finite field D-H) | 384 | Optional | RFC 3526 |
| 0004 0012h | 8 192-bit MODP group (finite field D-H) | 1 024 | Optional | RFC 3526 |
| 0004 0013h | 256-bit prime elliptic curve field P-256 | 32 | Optional | RFC 4753 |
| 0004 0015h | 521-bit prime elliptic curve field P-521 | 66 | Optional | RFC 4753 |
| 0004 0400h – 0004 FFFFh | Vendor Specific | | | |
| 0000 0000h – 0000 FFFFh | Restricted | | | IANA |
| All others | Reserved | | | |

The key length of the public value transferred in the KEY EXCHANGE DATA field (see 7.7.3.5.3) is determined by the algorithm identifier as shown in table x29.

The device server shall not include D-H IKEv2-SCSI cryptographic algorithm descriptors in the IKEv2-SCSI SA Creation Capabilities payload (see 7.7.3.5.12) fail to meet the following constraints:

a) Diffie-Hellman group sizes shall be 2 048 bits or larger; and
b) Elliptic curve field sizes shall be 256 bits or larger.

A Key Exchange step SECURITY PROTOCOL OUT command (see 5.13.4.7.2) shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to SA CREATION PARAMETER VALUE INVALID, if the parameter list contains:

a) No D-H IKEv2-SCSI cryptographic algorithm descriptors;
b) More than one D-H IKEv2-SCSI cryptographic algorithm descriptor; or
c) A D-H IKEv2-SCSI cryptographic algorithm descriptor that does not appear in the SA Creations Capabilities payload (see 7.7.3.5.12) most recently returned by the device server to any application client; or
d) An D-H IKEv2-SCSI cryptographic algorithm descriptor that contains an algorithm identifier that is not shown in table x29.

If a specific field is the cause for returning the SA CREATION PARAMETER VALUE INVALID additional sense code, then the SKSV bit shall be set to one and SENSE KEY SPECIFIC field shall be set as defined in 4.5.2.4.2.

The SA CREATION PARAMETER VALUE INVALID additional sense code indicates that the device server has abandoned the IKEv2-SCSI CCS (see 5.13.5).

{{SA CREATION PARAMETER VALUE INVALID is a new additional sense code.}}

#### 7.7.3.6.6 IKEv2-SCSI authentication algorithm (SA_AUTH) IKEv2-SCSI cryptographic algorithm descriptors

When the algorithm type field is set to SA_AUTH (i.e., F9h) in an IKEv2-SCSI cryptographic algorithm descriptor (see table x30), the descriptor specifies Authentication payload authentication algorithm used during the IKEv2-SCSI Authentication step (see 5.13.4.8).

**Table x30 — SA_AUTH IKEv2-SCSI cryptographic algorithm descriptor format**

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | ALGORITHM TYPE (F9h) | | | | | | | |
| 1 | Reserved | | | | | | | |
| 2 | (MSB) | | | IKE DESCRIPTOR LENGTH (000Ch) | | | | |
| 3 | | | | | | | | (LSB) |
| 4 | (MSB) | | | ALGORITHM IDENTIFIER | | | | |
| 7 | | | | | | | | (LSB) |
| 8 | Reserved | | | | | | | |
| 9 | Reserved | | | | | | | |
| 11 | | | | | | | | |

The ALGORITHM TYPE field and IKE DESCRIPTOR LENGTH field are described in 7.7.3.6.1.

The ALGORITHM IDENTIFIER field (see table x31) specifies Authentication payload authentication algorithm to which the SA_AUTH IKEv2-SCSI cryptographic algorithm descriptor applies.

**Table x31 — SA_AUTH ALGORITHM IDENTIFIER field**

| Code | Description | Support | Reference |
|---|---|---|---|
| 00F9 0000h | SA_AUTH_NONE | See table 45 [b] | this subclause |
| 00F9 0001h | RSA Digital Signature [a] | | RFC 4306 |
| 00F9 0002h | Shared Key Message Integrity Code | | RFC 4306 |
| 00F9 0009h | ECDSA with SHA-256 on the P-256 curve [a] | | RFC 4754 |
| 00F9 000Bh | ECDSA with SHA-512 on the P-521 curve [a] | | RFC 4754 |
| 00F9 0400h – 00F9 FFFFh | Vendor Specific | | |
| 0000 0000h – 0000 FFFFh | Restricted | | IANA |
| All others | Reserved | | |

[a] Use of certificates with this digital signature authentication algorithm is optional.
[b] SA_AUTH algorithm identifier support requirements are based on the value stored in the USAGE_TYPE SA parameter (see 5.13.2.2).

SA_AUTH_NONE specifies the omission of the IKEv2-SCSI Authentication step (see 5.13.4.8) as follows:

a) The presence of an SA_AUTH IKEv2-SCSI cryptographic algorithm descriptor with the ALGORITHM IDENTIFIER field set to SA_AUTH_NONE is an SA Creation Capabilities payload (see 7.7.3.5.12) indicates that the device server is willing to negotiate the omission of the IKEv2-SCSI Authentication step; and
b) The presence of an SA_AUTH IKEv2-SCSI cryptographic algorithm descriptor with the ALGORITHM IDENTIFIER field set to SA_AUTH_NONE is an IKEv2-SCSI Cryptographic Algorithms payload (see 7.7.3.5.13) indicates the following based upon the command whose parameter data carries the payload:
   A) In the parameter list for a Key Exchange SECURITY PROTOCOL OUT command (see 5.13.4.7.2), SA_AUTH_NONE specifies that the application client believes the device server allows the IKEv2-SCSI Authentication step to be skipped and is requesting that this be the case; and
   B) In the parameter data for a Key Exchange SECURITY PROTOCOL IN command (see 5.13.4.7.3), SA_AUTH_NONE indicates that the device server has agreed to skip the IKEv2-SCSI Authentication step.

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.

An SA_AUTH IKEv2-SCSI cryptographic algorithm descriptor with the ALGORITHM IDENTIFIER field set to SA_AUTH_NONE shall not appear in an IKEv2-SCSI Cryptographic Algorithms payload except under the circumstances described in 5.13.4.5.

The Shared Key Message Integrity Code is based on a shared secret that is associated with the identity in the Identification payload (see 7.7.3.5.4).

A USE bit set to one indicates that the device server is capable of authenticating itself using the Authentication payload authentication algorithm to which the SA_AUTH IKEv2-SCSI cryptographic algorithm descriptor applies. A USE bit set to zero indicates that the device server is not capable of authenticating itself using the Authentication payload authentication algorithm to which the SA_AUTH IKEv2-SCSI cryptographic algorithm descriptor applies. The USE bit shall be set to one in any SA_AUTH IKEv2-SCSI cryptographic algorithm descriptor where the algorithm identifier field contains SA_AUTH_NULL. The device server may ignore the USE bit in any IKEv2-SCSI Cryptographic Algorithms payload it receives.

The ACCEPT bit set to one indicates that the device server is capable of validating an application client authentication that uses the Authentication payload authentication algorithm to which the SA_AUTH IKEv2-SCSI cryptographic algorithm descriptor applies. The ACCEPT bit set to zero indicates that the device server is not capable of validating an application client authentication that uses the Authentication payload authentication algorithm to which the SA_AUTH IKEv2-SCSI cryptographic algorithm descriptor applies. The ACCEPT bit shall be set to one in any SA_AUTH IKEv2-SCSI cryptographic algorithm descriptor where the algorithm identifier field contains SA_AUTH_NULL. The device server may ignore the ACCEPT bit in any IKEv2-SCSI Cryptographic Algorithms payload it receives.

A Key Exchange step SECURITY PROTOCOL OUT command (see 5.13.4.7.2) shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to SA CREATION PARAMETER VALUE INVALID, if the parameter list contains:

a) No SA_AUTH IKEv2-SCSI cryptographic algorithm descriptors;
b) More than one SA_AUTH IKEv2-SCSI cryptographic algorithm descriptor; or
c) An SA_AUTH IKEv2-SCSI cryptographic algorithm descriptor that does not appear in the SA Creations Capabilities payload (see 7.7.3.5.12) most recently returned by the device server to any application client; or
d) An SA_AUTH IKEv2-SCSI cryptographic algorithm descriptor that contains an algorithm identifier that is not shown in table x31.

If a specific field is the cause for returning the SA CREATION PARAMETER VALUE INVALID additional sense code, then the SKSV bit shall be set to one and SENSE KEY SPECIFIC field shall be set as defined in 4.5.2.4.2.

The SA CREATION PARAMETER VALUE INVALID additional sense code indicates that the device server has abandoned the IKEv2-SCSI CCS (see 5.13.5).

{{SA CREATION PARAMETER VALUE INVALID is a new additional sense code.}}

### 7.7.3.7 Translating IKEv2 errors

IKEv2 (see RFC 4306) defines an error reporting mechanism based on the Notify payload. This standard translates such error reports into the device server and application actions defined in this subclause.

If a device server is required by IKEv2 to report an error using a Notify payload, the device server shall translate error into a CHECK CONDITION status with the sense key and additional sense code shown in table x32. The device server shall terminate the SECURITY PROTOCOL OUT command (see 6.30) that transferred the parameter list in which the IKE requirements for one or more payloads (see 7.7.3.5) require use of the Notify payload to report an error. The SECURITY PROTOCOL OUT command shall be terminated as described in this subclause. The device server shall not report the IKE errors described in this subclause by terminating a SECURITY PROTOCOL IN command (see 6.29) with a CHECK CONDITION status.

**Table x32 — IKEv2 Notify payload error translations for IKEv2-SCSI**

| IKEv2 (see RFC 4306) | | IKEv2-SCSI | |
|---|---|---|---|
| **Error Type** | **Description** | **Additional sense code** | **Sense key** |
| 0000h | Reserved | | |
| 0001h | UNSUPPORTED_CRITICAL_PAYLOAD | SA CREATION PARAMETER NOT SUPPORTED | ILLEGAL REQUEST |
| 0004h | INVALID_IKE_SPI | SA CREATION PARAMETER VALUE INVALID | |
| 0005h | INVALID_MAJOR_VERSION | | |
| 0007h | INVALID_SYNTAX [a] | | |
| 0009h | INVALID_MESSAGE_ID | | |
| 000Bh | INVALID_SPI | SA CREATION PARAMETER VALUE INVALID [b] | |
| 000Eh | NO_PROPOSAL_CHOSEN [k)] | SA CREATION PARAMETER VALUE INVALID | |
| 0011h | INVALID_KE_PAYLOAD [k)] | | |
| 0018h | AUTHENTICATION_FAILED | AUTHENTICATION FAILED | ABORTED COMMAND |
| 0022h - 0027h | See RFC 4306 [d] | n/a | n/a |
| 2000h - 3FFFh | Vendor Specific | | |
| All others | Restricted | | |

[a] This sense key and additional sense code shall be returned for a syntax error within an Encrypted payload (see 7.7.3.5.11) regardless of IKEv2 requirements to the contrary.

[b] SA CREATION 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 standard (see 3.1.17).

[c] An application client recovers by restarting processing with the Device Capabilities step (see 5.13.4.6) 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.

{{All additional sense codes in table x32 are new.}}

If an application client detects an IKEv2 error that RFC 4306 requires to be reported with a Notify payload, the application client should notify the device server by deleting the SA (see 5.13.6).

{{New parameter data subclause text ends here. Additions/deletions markups resume.}}

…

### 8.3.3.1 ACCESS CONTROL OUT introduction

The service actions of the ACCESS CONTROL OUT command (see table 420) are used to request service actions by the access controls coordinator to limit or grant access to the logical units by initiator ports. If the ACCESS CONTROL OUT command is implemented, the ACCESS CONTROL IN command also shall be implemented. The ACCESS CONTROL OUT command shall not be affected by access controls.

**Table 420 — ACCESS CONTROL OUT service actions**
**{{no changes in table 420 contents}}**

The ACCESS CONTROL OUT command may be addressed to any logical unit whose standard INQUIRY data (see 6.4.2) has the ACC bit set to one (e.g., LUN 0), in which case it shall be processed in the same manner as if the command had been addressed to the ACCESS CONTROLS well known logical unit. If an ACCESS CONTROL OUT command is received by a device server whose standard INQUIRY data has the ACC bit set to zero, the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID COMMAND OPERATION CODE.

If an ACCESS CONTROL OUT command is received while an IKEv2-SCSI CCS is in progress (see 5.13.4), the command shall be terminated with a CHECK CONDITION status, with the sense key NOT READY, and the additional sense code set to LOGICAL UNIT NOT READY, SA CREATION IN PROGRESS. The sense key specific additional sense data may be set as described in 5.13.7.

{{LOGICAL UNIT NOT READY, SA CREATION IN PROGRESS is a new additional sense code.}}

… {{The remainder of 8.3.3.1 is unchanged by this proposal.}} …

# Annex C
(Informative)
# IKEv2 protocol details and variations for IKEv2-SCSI

{{All of Annex C is new. Additions/deletions markups are not applied in this annex.}}

The IKEv2 protocol details and variations specified in RFC 4306 apply to IKEv2-SCSI (i.e., this standard) as follows:

a) Any SECURITY PROTOCOL IN command with an allocation length of up to 16 384 bytes is 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 is not terminated with an error due to the number of bytes transferred;
c) The timeout and retransmission mechanisms defined in RFC 4306 are not used by this standard (i.e., retransmission is performed by the applicable SCSI transport protocol);
d) Each SCSI command used by this standard completes by conveying a status from the device server to the application client;
e) This standard uses the MESSAGE ID field for sequencing (see RFC 4306), but only for SA creation (see 7.7.3.1);
f) The IKEv2 header EXCHANGE TYPE field is reserved in this standard because equivalent information is transferred in the SECURITY PROTOCOL OUT command and SECURITY PROTOCOL IN command CDBs;
g) The IKEv2 header VERSION bit is reserved in this standard;
h) This standard uses the pseudo-random functions (PRF) functions defined by RFC 4306;
i) The key derivation functions defined and used by this standard (see 5.13.3) are equivalent to the PRF+ found in RFC 4306;
j) The SA creation transactions (see 3.1.o) defined by this standard are not overlapped. If an application client attempts to start a second SA creation transaction before the first is completed, the offending command is terminated with CHECK CONDITION status (see 5.13.4.1), but this does not affect the SA creation transaction that is already in progress;
k) The NO_PROPOSAL_CHOSEN and INVALID_KE_PAYLOAD notify error types are replaced by the SA CREATION PARAMETER VALUE INVALID additional sense code (see 7.7.3.7) 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 5.13.4.7). An error can only occur if the application client has made an invalid selection, hence the SA CREATION PARAMETER VALUE INVALID description;
l) IKEv2 version numbers (see RFC 4306) are used by this standard (see 7.7.3.4), but the ability to respond to an unsupported version number with the highest version number to be used is not supported, and this standard does not include checks for version downgrade attacks;
m) IKEv2 cookies (see RFC 4306) are not used by this standard;
n) IKEv2 cryptographic algorithm negotiation (see RFC 4306) is replaced by the Device Server Capabilities step (see 5.13.4.6) and the Key Exchange step (see 5.13.4.7) (i.e., the IKEv2 proposal construct is not used by this standard);
o) In this standard an SA is rekeyed by replacing it with a new SA:
    A) CHILD_SAs are not used by this standard;
    B) The RFC 4306 discussion of CHILD_SAs does not apply to this standard;
    C) Coexistence of the original SA and the new SA that is achieved for rekeying purposes by restricting the device server's ability to delete SAs to the following cases:
        a) Expiration of a timeout (see 7.7.3.5.14);

b) Processing of an IKEv2-SCSI Delete function (see 5.13.6); and
c) Responding to an initial contact notification (see 7.7.3.5.8);
and

D) IKEv2 does not support rekeying notification for IKE_SAs, therefore this standard does not support rekeying notification;

p) The usage of Certificate Encodings in the Certificate payload and Certificate Request payload (see 7.7.3.5.5) are constrained as follows:
   A) In accordance with the recommendations in RFC 4718, Certificate Encoding values 01h-03h and 05h-0Ah are prohibited;
   B) This standard forbids the use URL-based Certificate Encodings (i.e., Certificate Encodings values 0Ch and 0Dh); and
   C) Certificate Encoding values that RFC 4306 defines as vendor specific are reserved in this standard;
q) Deleting an SA requires knowing the SAIs (i.e., SPIs) in both directions and including both SAIs in the Delete payload. The RFC 4306 description of the Delete payload is vague enough to allow this. The requirement is consistent with the SCSI model for SAs;
r) Traffic Selectors (see RFC 4306) are not used by this standard;
s) The requirements in RFC 4306 on nonces are be followed for the random nonces (see 3.1.95) defined by this standard;
t) The RFC 4306 requirements on address and port agility are specific to the user datagram protocol and the IP protocol and do not apply to this standard;
u) Keys for the Authentication step are generated as specified in RFC 4306;
v) This standard uses a slightly modified version of the authentication calculations in RFC 4306 (see 7.7.3.5.6);
w) The RFC 4306 sections that describe the following features are not used by this standard:
   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
x) IKEv2 Error Handling (see RFC 4306) is replaced by the use of CHECK CONDITION status and sense data by this standard. See 7.7.3.7 for details of how errors reported in the Notify payload are translated to sense data.

Where this standard uses IKE payload names (see 7.7.3.4) RFC 4306 uses the shorthand notation shown in table C.1.

**Table C.1 — IKE payload names shorthand**

| IKE payload name in this standard [a] | RFC 4306 shorthand [b] |
|---|---|
| Security Association | SAi or SAr |
| Key Exchange | KEi or KEr |
| Identification – Application Client | IDi |
| Identification – Device Server | IDr |
| Certificate | CERTi or CERTr |
| Certificate Request | CERTREQi |
| Authentication | AUTHi or AUTHr |
| Nonce | NONCEi or NONCEr |
| Notify | N-ICi or N-ICr |
| Delete | Di |
| Vendor ID | Vi or Vr |
| Traffic Selector – Application Client | TSi |
| Traffic Selector – Device Server | TSr |
| Encrypted | Ei or Er |
| Configuration | CPi or CPr |
| Extensible Authentication | EAPi or EAPr |

[a] To facilitate future enhancements, all IKE payloads are listed in this table, but not all entries in this table are used in this standard.

[b] In RFC 4306 the lowercase i indicates initiator and r indicates receiver. In this standard, the initiator is the application client and all such IKE payloads (e.g., KEi) appear in a SECURITY PROTOCOL OUT parameter list. The receiver is always the device server in this standard and all such IKE payloads (e.g., AUTHr) appear in SECURITY PROTOCOL IN parameter data.

{{The proposed SPC-4 changes end here.}}

## Summary of new additional sense codes

recommend 04h/13h for — LOGICAL UNIT NOT READY, SA CREATION IN PROGRESS
recommend 00h/1Eh for — CONFLICTING SA CREATION REQUEST
recommend 74h/0Ch for — UNABLE TO DECRYPT PARAMETER LIST
recommend 74h/10h for — SA CREATION PARAMETER VALUE INVALID
recommend 74h/30h for — SA CREATION PARAMETER NOT SUPPORTED
recommend 74h/40h for — AUTHENTICATION FAILED