ENDL
TEXAS

Date: 14 January 2008
To: T10 Technical Committee
From: Ralph O. Weber
Subject: SA Creation corrections and clarifications

Introduction

Revision History

r0 Initial revision
r1 Insert correct SPC-4 r12 subclause, table, and note numbers. Add changes to Encrypted payload processing based on suggestions from David Black. Cleanup cross references in the USAGE_DATA and MGMT_DATA SA parameter setup. Completed the initialization vector size changes.

Changes between r0 and r1 are marked with change bars.

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

Proposed Changes in SPC-4

{{Sorry. The first change is too big to fit on this page.}}

18484 Preston Road, Suite 102 PMB 178, Dallas, TX 75252 214-912-1373 972-596-2775 Fx ENDL_TX@computer.org

Change 1 – SK_xx key names too narrowly defined

{{Several of the SK_xx key names have meanings beyond those applied in IKEv2-SCSI SA creation. The following minimal changes should be made to reflect this in 5.13.4.4. A more complex change would be to add a second table for the SA key names, but this might not be helpful because the SK_xx names are the same in both cases.}}

Table 54 — IKEv2-SCSI shared key and SA shared key names

Name	Description	SA parameter (see 3.1.113) that stores this shared key
Shared keys used only during Authentication step		
SK_pi	The shared key used to construct the Authentication payload (see 7.6.3.5.6) for the SECURITY PROTOCOL OUT parameter list in the Authentication step (see 5.13.4.9.2).	shall not be stored in any SA parameter
SK_pr	The shared key used to construct the Authentication payload for the SECURITY PROTOCOL IN parameter data in the Authentication step.	
Shared keys used during IKEv2-SCSI SA creation and management		
SK_ai [a]	The shared key used to integrity check the Encrypted payload in the SECURITY PROTOCOL OUT parameter list in the: a) Authentication step; and b) IKEv2-SCSI Delete operation (see 5.13.4.13).	MGMT_DATA
SK_ar [a]	The shared key used to integrity check the Encrypted payload (see 7.6.3.5.10) in the SECURITY PROTOCOL IN parameter data in the Authentication step (see 5.13.4.9.3).	
SK_ei [a]	The shared key used to encrypt the Encrypted payload in the SECURITY PROTOCOL OUT parameter list in the: a) Authentication step; and b) IKEv2-SCSI Delete operation.	
SK_er [a]	The shared key used to encrypt the Encrypted payload in the SECURITY PROTOCOL IN parameter data in the Authentication step.	
Shared key used to construct the SA keys (see 3.1.112)		
SK_d	The shared key material that is used as input to the KDF that generates the KEYMAT SA parameter (see 3.1.113) bytes for the SA.	KEY_SEED

[a] The SA shared keys (see 3.1.112) SK_ai, SK_ar, SK_ei, and SK_er are stored in the KEYMAT SA parameter. SK_ai is intended to be used for integrity checking data in a Data-Out Buffer. SK_ar is intended to be used for integrity checking data in a Data-In Buffer. SK_ei is intended to be used for encrypting data in a Data-Out Buffer. SK_er is intended to be used for encrypting data in a Data-In Buffer.

Change 2 – Use Header DS SAI, not cryptographic algorithms

5.13.4.8.3 Key Exchange step SECURITY PROTOCOL IN command

…

If the Key Exchange step SECURITY PROTOCOL IN command (see 5.13.4.8.3) completes with GOOD status, the application client should copy the device server's SAI from the IKE_SA DEVICE SERVER SAI field in the IKEv2-SCSI header ~~SAI field in the IKEv2-SCSI SA Cryptographic Algorithms payload~~ to the state it is maintaining for the IKEv2-SCSI CCS.

Except for the SAI field, the ~~The~~ application client should compare the ~~other~~ fields in the IKEv2-SCSI SA Cryptographic Algorithms payload and the IKEv2-SCSI SAUT Cryptographic Algorithms payload, if any, to the values sent in the Key Exchange step SECURITY PROTOCOL OUT command (see 5.13.4.8.2). If the application client detects differences in the contents of the payloads other than in the SAI field, the application client should abandon the IKEv2-SCSI CCS and notify the device server that the IKEv2-SCSI CCS is being abandoned as described in 5.13.4.12.

…

5.13.4.9.3 Authentication step SECURITY PROTOCOL IN command

…

The application client should compare the ~~other~~ fields in the IKEv2-SCSI SAUT Cryptographic Algorithms payload to the values sent in the Authentication step SECURITY PROTOCOL OUT command (see 5.13.4.9.2). If the application client detects differences in the contents of the payloads ~~other than in the SAI field~~, the application client should abandon the IKEv2-SCSI CCS and notify the device server that the IKEv2-SCSI CCS is being abandoned as described in 5.13.4.12.

{{This change assumes that by the time of the Authentication step the application client should be expected to populate the Cryptographic Algorithms payload with the right SAI.}}

Change 3 – Clarify Usage Data

5.13.2.2 SA parameters

…

The USAGE_TYPE SA parameter shall be one of the values shown in table 48.

Table 48 — USAGE_TYPE SA parameter values

Value	Description	Usage data description	Reference
0000h - 0080h	Reserved		
0081h	Tape Data Encryption	None [a]	SSC-3
0082h - FFFFh	Reserved		

[a] The usage data length field in the IKEv2-SCSI SAUT Cryptographic Algorithms payload (see 7.6.3.5.13) shall contain zero.

{{The intent is that the Usage data description column contain 'see x.x.x' if the usage data format is defined in SPC-x or 'see UAS-x' if the usage data format is defined in another standard.}}

…

5.13.4.11 IKEv2-SCSI SA generation

…

K) USAGE_DATA shall contain at least the following values from of the usage data field in the IKEv2-SCSI SAUT Cryptographic Algorithms payload in the Key Exchange step SECURITY PROTOCOL OUT command:

~~a) The USAGE DATA field;~~

~~b) The ALGORITHM IDENTIFIER field (see 7.6.3.6) in the IKEv2-SCSI SAUT Cryptographic Algorithm descriptor (see 7.6.3.6) for the ENCR algorithm type;~~

~~c) The KEY LENGTH field (see 7.6.3.6.2) in the ALGORITHM ATTRIBUTES field in the IKEv2-SCSI SAUT Cryptographic Algorithm descriptor for the ENCR algorithm type; and~~

~~d) The ALGORITHM IDENTIFIER field (see 7.6.3.6) in the IKEv2-SCSI SAUT Cryptographic Algorithm descriptor for the INTEG algorithm type;~~

a) From the IKEv2-SCSI SAUT Cryptographic Algorithms payload (see 7.6.3.5.13) in the Key Exchange step or Authentication step SECURITY PROTOCOL OUT command, whichever applies:

A) The ALGORITHM IDENTIFIER field and KEY LENGTH field from the ENCR IKEv2-SCSI cryptographic algorithm descriptor (see 7.6.3.6.2); and

B) The ALGORITHM IDENTIFIER field from the INTEG IKEv2-SCSI cryptographic algorithm descriptor (see 7.6.3.6.4);

and

b) The contents, if any, of the USAGE DATA field;

{{The following change is made gratuitously to keep the text above and below this note in synch.}}

L) MGMT_DATA shall contain at least the following values:

a) From the IKEv2-SCSI SA Cryptographic Algorithms payload (see 7.6.3.5.12) in the Key Exchange step SECURITY PROTOCOL OUT command:

~~A) The ALGORITHM IDENTIFIER field (see 7.6.3.6) in the IKEv2-SCSI Cryptographic Algorithm descriptor (see 7.6.3.6) for the ENCR algorithm type;~~

~~B) The KEY LENGTH field (see 7.6.3.6.2) in the ALGORITHM ATTRIBUTES field in the IKEv2-SCSI Cryptographic Algorithm descriptor for the ENCR algorithm type;~~

~~C) The ALGORITHM IDENTIFIER field (see 7.6.3.6) in the IKEv2-SCSI Cryptographic Algorithm descriptor for the INTEG algorithm type;~~

A) The ALGORITHM IDENTIFIER field and KEY LENGTH field from the ENCR IKEv2-SCSI cryptographic algorithm descriptor (see 7.6.3.6.2); and

B) The ALGORITHM IDENTIFIER field from the INTEG IKEv2-SCSI cryptographic algorithm descriptor (see 7.6.3.6.4);

…

7.6.3.5.13 IKEv2-SCSI SAUT Cryptographic Algorithms payload

…

The SA TYPE field specifies the usage type for the SA and is selected from among those listed in table 48 (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 the error shall be processed as described in 7.6.3.8.3.

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.

~~NOTE 68 - 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 contents appear to be consistent with most SCSI length fields, now.}}

The size and format of the usage data depends on the SA type (see table 48 in 5.13.2.2). If the device server receives a USAGE DATA LENGTH field that contains a value that is inconsistent with the SA type, then the error shall be processed as described in 7.6.3.8.3.

The USAGE DATA field contains information to be stored in the USAGE_DATA SA parameter (see 3.1.113) if the SA is generated (see 5.13.4.11).

Change 4 – Use 4-byte SAIs throughout IKEv2-SCSI and ESP-SCSI

7.6.3.4 IKEv2-SCSI parameter data format

Table 361 shows the parameter list format used by 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., 41h).

Table 361 — 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)~~
0				Restricted (see RFC 4306)				
3								
4	(MSB)			IKE_SA APPLICATION CLIENT SAI				
7								(LSB)
8				Restricted (see RFC 4306)				
11								
12	(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				IKEv2-SCSI payload [first] (see 7.6.3.5)				
	. . .							
				IKEv2-SCSI payload [last] (see 7.6.3.5)				
n								

…

7.6.3.5.8 Notify payload

This standard uses the Notify payload (see table 373) only to provide initial contact notification from the application client to the device server.

Table 373 — Notify payload format

Bit / Byte	7	6	5	4	3	2	1	0
0	NEXT PAYLOAD							
1	CRIT (1b)	Reserved						
2	(MSB)	IKE PAYLOAD LENGTH (0010h)						
3								(LSB)
4	PROTOCOL ID (01h)							
5	SAI SIZE (08h)							
6	(MSB)	NOTIFY MESSAGE TYPE (4000h)						
7								(LSB)
~~8~~	~~SAI~~							
~~15~~								
8	Restricted (see RFC 4306)							
11								
12	(MSB)	SAI						
15								(LSB)

{{The other choice is to reduce the value in the SAI SIZE field to four.}}

The NEXT PAYLOAD field, CRIT bit, and IKE PAYLOAD LENGTH field are described in 7.6.3.5.1.

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 field set to a value other than one, then the IKEv2-SCSI CCS state maintained for the I_T_L nexus shall be abandoned as described in 7.6.3.8.3.

The SAI SIZE field contains eight. If the device server receives a value other than eight in the SAI SIZE field, then the IKEv2-SCSI CCS state maintained for the I_T_L nexus shall be abandoned as described in 7.6.3.8.3.

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 the IKEv2-SCSI CCS state maintained for the I_T_L nexus shall be abandoned as described in 7.6.3.8.3.

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 7.6.3.4), then the IKEv2-SCSI CCS state maintained for the I_T_L nexus shall be abandoned as described in 7.6.3.8.3.

…

7.6.3.5.9 Delete payload

The Delete payload (see table 374) requests the deletion of an existing SA or the abandonment of an IKEv2-SCSI CCS that is in progress.

Table 374 — Delete payload format

Bit / Byte	7	6	5	4	3	2	1	0
0	NEXT PAYLOAD							
1	CRIT (1b)	Reserved						
2	(MSB)	IKE PAYLOAD LENGTH (0018h)						
3								(LSB)
4	PROTOCOL ID (01h)							
5	SAI SIZE (08h)							
6	(MSB)	NUMBER OF SAIS (0002h)						
7								(LSB)
~~8~~	AC_SAI							
~~15~~								
~~16~~	DS_SAI							
~~23~~								
8	Restricted (see RFC 4306)							
11								
12	(MSB)	AC_SAI						
15								(LSB)
16	Restricted (see RFC 4306)							
19								
20	(MSB)	DS_SAI						
23								(LSB)

{{The other choice is to reduce the value in the SAI SIZE field to four.}}

The NEXT PAYLOAD field, CRIT bit, and IKE PAYLOAD LENGTH field are described in 7.6.3.5.1.

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 field set to a value other than one, then the error shall be processed as described in 7.6.3.8.3.

The SAI SIZE field contains eight. If the device server receives a value other than eight in the SAI SIZE field, then the error shall be processed as described in 7.6.3.8.3.

The AC_SAI field contains the AC_SAI SA parameter value (see 3.1.113) for the SA to be deleted. If the contents of the AC_SAI field do not match the contents of the IKE_SA APPLICATION CLIENT SAI field in the IKEv2-SCSI header (see 7.6.3.4), then the error shall be processed as described in 7.6.3.8.3.

The DS_SAI field contains the DS_SAI SA parameter value (see 3.1.113) for the SA to be deleted. If the contents of the DS_SAI field do not match the contents of the IKE_SA DEVICE SERVER SAI field in the IKEv2-SCSI header (see 7.6.3.4), then the error shall be processed as described in 7.6.3.8.3.

…

7.6.3.5.12 IKEv2-SCSI SA Cryptographic Algorithms payload

The IKEv2-SCSI SA Cryptographic Algorithms payload (see table 378) lists the security algorithms that are being used in the creation and management (e.g., deletion) of an SA using an IKEv2-SCSI CCS.

Table 378 — IKEv2-SCSI SA Cryptographic Algorithms payload format

Bit / Byte	7	6	5	4	3	2	1	0
0	NEXT PAYLOAD							
1	CRIT (1b)	Reserved						
2	(MSB)	IKE PAYLOAD LENGTH (n+1)						
3								(LSB)
4		Reserved						
5								
6	(MSB)	USAGE DATA LENGTH (0000h)						
7								(LSB)
8	(MSB)	SAI						
15								(LSB)
16		Reserved						
18								
19	NUMBER OF ALGORITHM DESCRIPTORS							
IKEv2-SCSI cryptographic algorithm descriptors								
20		IKEv2-SCSI cryptographic algorithm descriptor [first] (see 7.6.3.6)						
		. . .						
n		IKEv2-SCSI cryptographic algorithm descriptor [last] (see 7.6.3.6)						

The NEXT PAYLOAD field, CRIT bit, and IKE PAYLOAD LENGTH field are described in 7.6.3.5.1.

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

The USAGE DATA LENGTH field is set to zero in the IKEv2-SCSI SA Cryptographic Algorithms payload.

The SAI field is reserved.

{{No changes are suggested because the SAI field is reserved.}}

…

7.6.3.5.13 IKEv2-SCSI SAUT Cryptographic Algorithms payload

The IKEv2-SCSI SAUT Cryptographic Algorithms payload (see table 379) lists the usage type of and security algorithms to be used by the SA that is created as a result of an IKEv2-SCSI CCS.

Table 379 — IKEv2-SCSI SAUT Cryptographic Algorithms payload format

Bit Byte	7	6	5	4	3	2	1	0
0	NEXT PAYLOAD							
1	CRIT (1b)	Reserved						
2	(MSB)	IKE PAYLOAD LENGTH (n+1)						
3								(LSB)
4	(MSB)	SAI						
11								(LSB)
12	(MSB)	SA TYPE						
13								(LSB)
14	(MSB)	USAGE DATA LENGTH (j)						
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		IKEv2-SCSI cryptographic algorithm descriptor [first] (see 7.6.3.6)						
		. . .						
n		IKEv2-SCSI cryptographic algorithm descriptor [last] (see 7.6.3.6)						

The NEXT PAYLOAD field, CRIT bit, and IKE PAYLOAD LENGTH field are described in 7.6.3.5.1.

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

The SAI field is reserved.

{{No changes are suggested because the SAI field is reserved.}}

…

Change 5 – Multiple Certificates Are Good

7.6.3.5.5 Certificate payload and Certificate Request payload

…

The relationship between the Certificate payload and the Identification payload is described in 7.6.3.5.4.

Device servers that support certificates should support a mechanism outside the scope of this standard for replacing certificates and have the ability to store more than one certificate to facilitate such replacements.

Change 6 – How to hash Authentication payload RSAs

7.6.3.6.6 IKEv2-SCSI authentication algorithm IKEv2-SCSI cryptographic algorithm descriptors

…

The ALGORITHM IDENTIFIER field (see table 392) specifies Authentication payload authentication algorithm to which the SA_AUTH_OUT IKEv2-SCSI cryptographic algorithm descriptor or SA_AUTH_IN IKEv2-SCSI cryptographic algorithm descriptor applies.

Table 392 — SA_AUTH_OUT and SA_AUTH_IN ALGORITHM IDENTIFIER field

Code	Description	Support	Reference
00F9 0000h	SA_AUTH_NONE	Optional	this subclause
00F9 0001h	RSA Digital Signature with SHA-1 [a]	Optional	RFC 4306
00F9 0002h	Shared Key Message Integrity Code	Optional	RFC 4306 [b, c]
00F9 0009h	ECDSA with SHA-256 on the P-256 curve [a]	Optional	RFC 4754
00F9 000Bh	ECDSA with SHA-512 on the P-521 curve [a]	Optional	RFC 4754
00F9 00C9h – 00F9 00FFh	Vendor Specific	Optional	
0000 0000h – 0000 FFFFh	Restricted	Prohibited	IANA
All others	Reserved		

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

[b] The 17 ASCII character non-terminated pre-shared key (see 3.1.94) pad string "Key Pad for IKEv2" specified by RFC 4306 is replaced by the 22 ASCII character non-terminated pre-shared key pad string "Key Pad for IKEv2-SCSI".

[c] The pre-shared key (see 3.1.94) requirements used by this standard (see 5.13.4.5) apply in addition to those found in RFC 4306.

Change 7 – 'this subclause'?

7.6.3.7 Errors in IKEv2-SCSI security protocol commands

For a single I_T_L nexus, the device server shall ensure that the two or four IKEv2-SCSI CCS commands are processed in the order described in 5.13.4.1 based only on the contents of the CDB (i.e., the SECURITY PROTOCOL OUT parameter data shall not be processed unless the tests in table 393 specify the processing of the command) using the tests and responses shown in table 393

Table 393 — IKEv2-SCSI command ordering processing requirements on a single I_T_L nexus

{{Big, hairy table whose contents are not relevant to the issue and that does not need to be changed.}}

The processing shown in table 393 shall be performed before the parameter data ~~any other~~ error handling described in 7.6.3.8 ~~this subclause~~.

7.6.3.8 Errors in IKEv2-SCSI security protocol parameter data

7.6.3.8.1 Overview

Errors in the parameter data transferred to the device sever by an IKEv2-SCSI SECURITY PROTOCOL OUT command are classified (see table 393) based on the ease with which they may be used to mount denial of service attacks against IKEv2-SCSI SA creation operations by an attacker that has not participated as the application client or device server in the Key Exchange step SECURITY PROTOCOL OUT command (see 5.13.4.8.2) that started the IKEv2-SCSI CCS.

Change 8 – IKEv2-SCSI header definition is incomplete

{{The IKEv2-SCSI header definition does not cover the possibility of a header appearing in a Delete operation.}}

7.6.3.4 IKEv2-SCSI parameter data format

Table 361 shows the parameter list format used by 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., 41h).

Table 361 — IKEv2-SCSI SECURITY PROTOCOL OUT and SECURITY PROTOCOL IN parameter data

{{Big, hairy table whose contents are changed by Change 4, but not here.}}

The IKE_SA APPLICATION CLIENT SAI field contains the value that is or is destined to become the AC_SAI SA parameter (see 5.13.2.2) ~~when the~~ in the generated SA ~~is generated~~ (see 5.13.4.11). The AC_SAI is chosen by the application client to uniquely identify its representation of the SA that is being negotiated or managed (e.g., deleted).

If the device server receives an IKEv2-SCSI header with the IKE_SA APPLICATION CLIENT SAI field set to zero, then the error shall be processed as described in 7.6.3.8.

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 or SA management. If the two values are not identical, the application client should abandon the IKEv2-SCSI CCS, if any, and notify the device server that the IKEv2-SCSI CCS, if any, is being abandoned as described in 5.13.4.12.

Except in the Key Exchange step SECURITY PROTOCOL OUT command (see 5.13.4.8.2), the IKE_SA DEVICE SERVER SAI field contains the value that is or is destined to become the DS_SAI SA parameter ~~when the~~ in the generated SA ~~is generated~~. The DS_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 abandon the IKEv2-SCSI CCS and notify the device server that the IKEv2-SCSI CCS is being abandoned as described in 5.13.4.12.

The device server shall validate the contents of the IKE_SA APPLICATION CLIENT SAI field and the IKE_SA DEVICE SERVER SAI field as shown in table x1.

Table x1 — IKEv2-SCSI header checking of SAIs

Contents of SECURITY PROTOCOL SPECIFIC field in SECURITY PROTOCOL OUT CDB	Expected contents for …		Device server action if expected field contents not received
	IKE_SA APPLICATION CLIENT SAI field	IKE_SA DEVICE SERVER SAI FIELD	
0102h (i.e., Key Exchange step)	any value	reserved	No actions taken based on expected field contents
0103h (i.e., Authentication step)	A match with the SAI values maintained for an IKEv2-SCSI CCS on the I_T_L nexus on which the command was received		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 SA CREATION PARAMETER VALUE REJECTED; and b) The device server shall continue the IKEv2-SCSI CCS by preparing to receive another Authentication step SECURITY PROTOCOL OUT command
0104h (i.e., Delete operation)	1) A match with the SAI values maintained for an IKEv2-SCSI CCS on the I_T_L nexus on which the command was received		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 SA CREATION PARAMETER VALUE REJECTED; and b) The device server shall continue the IKEv2-SCSI CCS by preparing to receive another Authentication step SECURITY PROTOCOL OUT command
	2) A match with the SAI values maintained for any active SA		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 PARAMETER LIST

~~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 on the I_T_L nexus on which the command was received. 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 SA CREATION PARAMETER VALUE REJECTED; and~~

~~b) The device server shall continue the IKEv2-SCSI CCS by preparing to receive another Authentication step SECURITY PROTOCOL OUT command.~~

Change 9 – Provide the sizes of Initialization Vectors

7.6.3.5.10 Encrypted payload

7.6.3.5.10.1 Introduction

…

The INITIALIZATION VECTOR field contains the initialization vector encryption algorithm input value. The size of the initialization vector is defined by the encryption algorithm as shown in table 384 (see 7.6.3.6.2).

…

7.6.3.6.2 Encryption algorithm (ENCR) IKEv2-SCSI cryptographic algorithm descriptors

…

The ALGORITHM IDENTIFIER field (see table 384) specifies the encryption algorithm to which the ENCR IKEv2-SCSI cryptographic algorithm descriptor applies.

Table 384 — ENCR ALGORITHM IDENTIFIER field (Sheet 1 of 2)

Code	Description	Salt [a] length (bytes)	IV [b] length (bytes)	Key length (bytes)	Support	Reference
8001 000Bh	ENCR_NULL [c]	n/a		0	Mandatory	
8001 000Ch	AES-CBC [c]	n/a	16	16	Optional	RFC 3602
				24	Prohibited	
				32	Optional	
8001 0010h	AES-CCM with a 16 byte MAC [d]	3	8	16	Optional	RFC 4309
				24	Prohibited	
				32	Optional	
8001 0014h	AES-GCM with a 16 byte MAC [d]	4	8	16	Optional	RFC 4106
				24	Prohibited	
				32	Optional	
8001 0400h – 8001 FFFFh	Vendor Specific					

[a] See RFC 4106 and RFC 4309.

[b] Initialization Vector.

[c] If the INTEG cryptographic algorithm descriptor (see 7.6.3.6.4) in the same IKEv2-SCSI SA Cryptographic Algorithms payload or the same IKEv2-SCSI SAUT Cryptographic Algorithms payload as this ENCR cryptographic algorithm descriptor has the ALGORITHM IDENTIFIER field set to AUTH_COMBINED, then the error shall be processed as described in 7.6.3.8.3.

[d] If the INTEG cryptographic algorithm descriptor (see 7.6.3.6.4) in the same IKEv2-SCSI SA Cryptographic Algorithms payload or the same IKEv2-SCSI SAUT Cryptographic Algorithms payload as this ENCR cryptographic algorithm descriptor does not have the ALGORITHM IDENTIFIER field set to AUTH_COMBINED, then the error shall be processed as described in 7.6.3.8.3.

Table 384 — ENCR ALGORITHM IDENTIFIER field (Sheet 2 of 2)

Code	Description	Salt [a] length (bytes)	IV [b] length (bytes)	Key length (bytes)	Support	Reference
0000 0000h – 0000 FFFFh	Restricted					IANA
All others	Reserved					

[a] See RFC 4106 and RFC 4309.

[b] Initialization Vector.

[c] If the INTEG cryptographic algorithm descriptor (see 7.6.3.6.4) in the same IKEv2-SCSI SA Cryptographic Algorithms payload or the same IKEv2-SCSI SAUT Cryptographic Algorithms payload as this ENCR cryptographic algorithm descriptor has the ALGORITHM IDENTIFIER field set to AUTH_COMBINED, then the error shall be processed as described in 7.6.3.8.3.

[d] If the INTEG cryptographic algorithm descriptor (see 7.6.3.6.4) in the same IKEv2-SCSI SA Cryptographic Algorithms payload or the same IKEv2-SCSI SAUT Cryptographic Algorithms payload as this ENCR cryptographic algorithm descriptor does not have the ALGORITHM IDENTIFIER field set to AUTH_COMBINED, then the error shall be processed as described in 7.6.3.8.3.

Change 10 – The Encrypted payload must apply for disaster assistance

7.6.3.5.10 Encrypted payload

7.6.3.5.10.a Combined mode encryption

The following types of encryption algorithms are defined to construct and decode the Encrypted payload:

a) Non-combined modes that use separate algorithms to encrypt/decrypt and integrity check the Encrypted payload; and
b) Combined modes in which a single encryption algorithm does both encryption/decryption and integrity checking.

The ALGORITHM IDENTIFIER field in the INTEG IKEv2-SCSI cryptographic algorithm descriptor (see 7.6.3.6.4) in the IKEv2-SCSI SA Cryptographic Algorithms payload (see 7.6.3.5.12) in the Key Exchange step (see 5.13.4.8) indicates to type of encryption algorithm to be applied to the Encrypted payload for IKEv2-SCSI functions as follows:

a) If the ALGORITHM IDENTIFIER field is not set to AUTH_COMBINED, a non-combined mode encryption algorithm is being used; or
b) If the ALGORITHM IDENTIFIER field is set to AUTH_COMBINED, a combined mode encryption algorithm is being used.

If the Encrypted payload is in parameter data that is not associated with an active IKEv2-SCSI CCS (see 3.1.54), then the integrity checking algorithm identifier that selects between combined and non-combined mode encryption is found in the MGMT_DATA SA parameter (see 3.1.114).

7.6.3.5.10.1 Encryption payload Introduction

The Encrypted payload transfers (see table 375) one or more other IKEv2-SCSI payloads that are encrypted and integrity checked from the application client to the device server and vice versa.

If IKEv2-SCSI parameter data contains the Encrypted payload, then the Encrypted payload is the first payload in the parameter data (i.e., the NEXT PAYLOAD field in the IKEv2-SCSI header (see 7.6.3.5.1) contains 2Eh). Since the NEXT PAYLOAD field in an Encrypted payload identifies the first payload in the CIPHERTEXT field, there is no way to identify a payload following the Encrypted payload, and none are allowed.

Table 375 — Encrypted payload format

{{Just another boring payload structure format table that is not needed to understand the changes.}}

The NEXT PAYLOAD field identifies the first IKEv2-SCSI payload in the CIPHERTEXT field using the coded values shown in table 362 (see 7.6.3.5.1).

The CRIT bit and IKE PAYLOAD LENGTH field are described in 7.6.3.5.1.

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

The IKE PAYLOAD LENGTH field contains the number of bytes that follow in the Encrypted payload. The number of bytes in the CIPHERTEXT field is equal to the number of bytes in the plaintext (see table 376).

The INITIALIZATION VECTOR field contains the initialization vector encryption algorithm input value. The size of the initialization vector is defined by the encryption algorithm. {{Note: Change 9 modifies this text.}}

The CIPHERTEXT field contains the result of processing the encryption algorithm specified by the ALGORITHM IDENTIFIER field in the ENCR IKEv2-SCSI cryptographic algorithm descriptor (see 7.6.3.6.2) in the IKEv2-SCSI SA Cryptographic Algorithms payload (see 7.6.3.5.12) in the Key Exchange step (see 5.13.4.8) using the inputs:

a) IKEv2-SCSI AAD is an encryption algorithm input as follows:
   A) If a non-combined mode encryption algorithm is being used (see 7.6.3.5.10.a), then no AAD input is needed or provided; or
   B) If a combined mode encryption algorithm is being used (see 7.6.3.5.10.a), then the AAD described in 7.6.3.5.10.2 is an input;
b) A byte string composed of:
   ~~1) The AAD, if any, described in 7.6.3.5.10.2;~~
   1) The contents of the INITIALIZATION VECTOR field (see table 375); and
   2) The plaintext data shown in table 376;
   and
c) The shared key value, key length, and salt value (see table 384 in 7.6.3.6.2) if any, for ~~from~~ one of the following shared keys:
   A) If the Encrypted payload appears in the parameter data for an Authentication step SECURITY PROTOCOL OUT command (see 5.13.4.9.2) ~~parameter data~~, the SK_ei shared key (see 5.13.4.4) for the IKEv2-SCSI CCS (see 3.1.54); ~~or~~
   B) If the Encrypted payload appears in the parameter data for an Authentication step SECURITY PROTOCOL IN command (see 5.13.4.9.3) ~~parameter data~~, the SK_er shared key (see 5.13.4.4) for the IKEv2-SCSI CCS.; or
   C) If the Encrypted payload appears in the parameter data for a Delete operation SECURITY PROTOCOL OUT command (see 5.13.4.13), the SK_ei shared key (see 5.13.4.4) from the MGMT_DATA SA parameter (see 3.1.114 and 5.13.4.11).

NOTE x1 - Salt values (see table 384 in 7.6.3.6.2) are used only by combined mode encryption algorithms (see 7.6.3.5.10.a).

If the Encrypted payload appears in the parameter data for a Delete operation SECURITY PROTOCOL OUT command, then the encryption algorithm identifier stored in the MGMT_DATA SA parameter indicates the encryption algorithm to use.

The INTEGRITY CHECK VALUE field contains the integrity check value that is computed as described in 7.6.3.5.10.a. ~~output by integrity algorithm or encryption algorithm (i.e., if the integrity algorithm is AUTH_COMBINED the encryption algorithm includes integrity checking capabilities).~~ The size of the integrity check value is defined by the integrity algorithm or encryption algorithm, depending on which algorithm computes the value.

~~If the integrity algorithm specified by the ALGORITHM IDENTIFIER field in the INTEG IKEv2-SCSI cryptographic algorithm descriptor (see 7.6.3.6.4) in the IKEv2-SCSI SA Cryptographic Algorithms payload (see 7.6.3.5.12) in the Key Exchange step (see 5.13.4.8) is AUTH_COMBINED, then the inputs to and computation of the INTEGRITY CHECK VALUE field are the same as the inputs to and computation of the CIPHERTEXT field.~~

If the integrity algorithm specified by the ALGORITHM IDENTIFIER field in the INTEG IKEv2-SCSI cryptographic algorithm descriptor in the IKEv2-SCSI SA Cryptographic Algorithms payload in the Key Exchange step is not AUTH_COMBINED, then the contents of the INTEGRITY CHECK VALUE field are computed by processing the integrity check algorithm specified by the ALGORITHM IDENTIFIER field in the INTEG IKEv2-SCSI cryptographic algorithm descriptor using the following inputs:

a) A byte string composed of:
   1) The AAD~~, if any,~~ described in 7.6.3.5.10.2;
   2) The contents of the INITIALIZATION VECTOR field (see table 375); and
   3) The ~~plaintext~~ ciphertext that is the result of encrypting the plaintext data ~~shown in table 376~~;
   and
b) The shared key value from one of the following shared keys:
   A) If the Encrypted payload appears in the parameter data for an Authentication step SECURITY PROTOCOL OUT command (see 5.13.4.9.2) ~~parameter data~~, the SK_ai shared key (see 5.13.4.4) for the IKEv2-SCSI CCS (see 3.1.54); ~~or~~
   B) If the Encrypted payload appears in the parameter data for an Authentication step SECURITY PROTOCOL IN command (see 5.13.4.9.3) ~~parameter data~~, the SK_ar shared key (see 5.13.4.4) for the IKEv2-SCSI CCS~~.~~; or
   C) If the Encrypted payload appears in the parameter data for a Delete operation SECURITY PROTOCOL OUT command (see 5.13.4.13), the SK_ai shared key (see 5.13.4.4) from the MGMT_DATA SA parameter (see 3.1.114 and 5.13.4.11).

If the Encrypted payload appears in the parameter data for a Delete operation SECURITY PROTOCOL OUT command, then the integrity checking algorithm identifier stored in the MGMT_DATA SA parameter indicates the integrity checking algorithm to use.

When computing the encrypted CIPHERTEXT field contents for an Encrypted payload, the plaintext shown in table 376 is used.

Table 376 — Plaintext format for Encrypted payload CIPHERTEXT field

{{Just another boring payload structure format table that is not needed to understand the changes.}}

Each IKEv2-SCSI payload (see 7.6.3.5) contains specific data related to the operation being performed. A specific combination of IKEv2-SCSI payloads is needed for each operation (e.g., Authentication) as summarized in 5.13.4.2.

The PADDING BYTES field contains zero to 255 bytes. The number of padding bytes is:

a) Defined by the encryption algorithm; or
b) Any number of bytes that causes the length of all plaintext bytes (i.e., l+2) to be a whole multiple of the cipher block size for the encryption algorithm being used.

The contents of the padding bytes are:

a) Defined by the encryption algorithm; or
b) If the encryption algorithm does not define the padding bytes contents, a series of one byte binary values starting at one and incrementing by one in each successive byte (i.e., 01h in the first padding byte, 02h in the second padding byte, etc.).

If the encryption algorithm does not place requirements on the contents of the padding bytes (i.e., option b) is in effect), then after decryption the contents of the padding bytes shall be verified to match the series of one byte binary values described in this subclause. If this verification is not successful in a device server, the error shall be processed as described in 7.6.3.8.2. If this verification is not successful in an application client, the decrypted data should be ignored.

The PAD LENGTH field contains the number of bytes in the PADDING BYTES field.

7.6.3.5.10.2 IKEv2-SCSI AAD ~~for combined mode encryption algorithms~~

~~For IKEv2-SCSI, encryption algorithms that also provide integrity checking require AAD as an input to their encryption and decryption functions.~~ The AAD defined by this standard for IKEv2-SCSI use is as follows:

1) All the bytes in the IKEv2-SCSI Header (see 7.6.3.4); and
2) All the bytes in the IKEv2-SCSI Payload Header (see 7.6.3.5.1) of the Encrypted payload (see 7.6.3.5.10).

Encryption algorithms that do not provide integrity checking do not use AAD.

7.6.3.5.10.3 Processing a received Encrypted payload

{{This text assumes that the IKEv2-SCSI header validation has been performed as described in Change 8.}}

Before performing any checks of data contained in the CIPHERTEXT field, the contents of the INTEGRITY CHECK VALUE field and CIPHERTEXT field shall be integrity checked and 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 7.6.3.4) as described in this subclause.

Computation of the comparison integrity check value and decryption of an Encrypted payload is performed as follows:

1) If a non-combined mode encryption algorithm is being used (see 7.6.3.5.10.a), then the comparison integrity check value is computed by performing the integrity check algorithm specified by the ALGORITHM IDENTIFIER field in the INTEG IKEv2-SCSI cryptographic algorithm descriptor using the following inputs:
   A) A byte string composed of:
      1) The AAD described in 7.6.3.5.10.2;
      2) The contents of the INITIALIZATION VECTOR field (see table 375) in the Encrypted payload; and
      3) The contents of the ciphertext field (see table 375) in the Encrypted payload;
      and
   B) The shared key value from one of the following shared keys:
      a) If the Encrypted payload appears in the parameter data for an Authentication step SECURITY PROTOCOL OUT command (see 5.13.4.9.2), the SK_ai shared key (see 5.13.4.4) for the IKEv2-SCSI CCS (see 3.1.54);
      b) If the Encrypted payload appears in the parameter data for an Authentication step SECURITY PROTOCOL IN command (see 5.13.4.9.3), the SK_ar shared key (see 5.13.4.4) for the IKEv2-SCSI CCS; or

c) If the Encrypted payload appears in the parameter data for a Delete operation SECURITY PROTOCOL OUT command (see 5.13.4.14), the SK_ai shared key (see 5.13.4.4) from the MGMT_DATA SA parameter (see 3.1.114 and 5.13.4.11);

and

2) The plaintext data – and if a combined mode encryption algorithm is being used (see 7.6.3.5.10.a), the comparison integrity check value – are computed by performing the encryption algorithm specified by the ALGORITHM IDENTIFIER field in the ENCR IKEv2-SCSI cryptographic algorithm descriptor (see 7.6.3.6.2) in the IKEv2-SCSI SA Cryptographic Algorithms payload (see 7.6.3.5.12) in the Key Exchange step (see 5.13.4.8) using the inputs:
   A) A byte string composed of:
      1) The contents of the INITIALIZATION VECTOR field (see table 375) in the Encrypted payload; and
      2) The contents of the ciphertext field (see table 375) in the Encrypted payload;

      and
   B) The shared key value, key length, and salt value (see table 384 in 7.6.3.6.2) if any, for one of the following shared keys:
      a) If the Encrypted payload appears in the parameter data for an Authentication step SECURITY PROTOCOL OUT command (see 5.13.4.9.2), the SK_ei shared key (see 5.13.4.4) for the IKEv2-SCSI CCS (see 3.1.54);
      b) If the Encrypted payload appears in the parameter data for an Authentication step SECURITY PROTOCOL IN command (see 5.13.4.9.3), the SK_er shared key (see 5.13.4.4) for the IKEv2-SCSI CCS; or
      c) If the Encrypted payload appears in the parameter data for a Delete operation SECURITY PROTOCOL OUT command (see 5.13.4.14), the SK_ei shared key (see 5.13.4.4) from the MGMT_DATA SA parameter (see 3.1.114 and 5.13.4.11).

If the Encrypted payload appears in the parameter data for a Delete operation SECURITY PROTOCOL OUT command, then the integrity checking algorithm identifier value and encryption algorithm identifier value that are stored in the MGMT_DATA SA parameter indicate the integrity checking algorithm to use.

If the comparison integrity check value differs from the value in the INTEGRITY CHECK VALUE field of the Encrypted payload:

a) The application client should abandon the IKEv2-SCSI CCS and notify the device server that it is abandoning the IKEv2-SCSI CCS as described in 5.13.4.12; and
b) The device server shall respond to the mismatch as follows:
   A) If the IKEv2-SCSI header (see 7.6.3.4) specifies an attempt to provide authentication data for or the deletion of an IKE-v2-SCSI CCS on the I_T_L nexus on which the command was received, 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 SA CREATION PARAMETER VALUE REJECTED; and
      b) The device server shall continue the IKEv2-SCSI CCS by preparing to receive another Authentication step SECURITY PROTOCOL OUT command;

      or
   B) If the IKEv2-SCSI header (see 7.6.3.4) specifies the deletion of an active SA, 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 PARAMETER LIST.

{{All of the text that follows is marked for deletion. The text above this point should completely cover all the cases shown below. Reviewers are encouraged to verify this.}}

Device server processing of an Encrypted payload in a SECURITY PROTOCOL OUT command parameter list shall proceed as follows:

a) If the following are true:
   A) The SECURITY PROTOCOL SPECIFIC field in the CDB contains 0104h (i.e., Delete operation);
   B) The contents of the IKE_SA APPLICATION CLIENT SAI field match the AC_SAI SA parameter (see 3.1.114) in a generated SA for which IKEv2-SCSI CCS processing has been completed; and
   C) The IKE_SA DEVICE SERVER SAI field match the DS_SAI SA parameter in the same generated SA;
   then the contents of the MGMT_DATA SA parameter shall be used to decrypt and integrity check the Encrypted payload as described in 7.6.3.5.10.4;
b) If the following are true:
   A) If the device server is maintaining state for an IKEv2-SCSI CCS on the I_T_L nexus on which the command was received;
   B) The IKEv2-SCSI CCS has completed the Key Exchange step;
   C) The SECURITY PROTOCOL SPECIFIC field in the CDB contains 0103h (i.e., Authentication step) or 0104h (i.e., Delete operation);
   D) The contents of the IKE_SA APPLICATION CLIENT SAI field match the application client's SAI in the maintained IKEv2-SCSI CCS state; and
   E) The contents of the IKE_SA DEVICE SERVER SAI field match the device server's SAI in the maintained IKEv2-SCSI CCS state;
   then the payload shall be decrypted as described in 7.6.3.5.10.5; or
c) If the conditions described in a) and b) are not met or if an error is detected while decrypting or integrity checking the CIPHERTEXT field contents, then the error shall be processed as described in 7.6.3.8.2.

If an application client receives an Encrypted payload in a SECURITY PROTOCOL IN command parameter list, then the payload shall be decrypted and integrity checked as described in 7.6.3.5.10.6.

7.6.3.5.10.4 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 5.13.4.9.2) for a generated SA, the device server shall decrypt and check the integrity of the Encrypted payload as follows:

1) Decrypt the CIPHERTEXT field of the Encrypted payload using the contents of the INITIALIZATION VECTOR field in the Encrypted payload (see 7.6.3.5.10.1), the AAD described in 7.6.3.5.10.2, and following information from the MGMT_DATA_SA parameter (see 5.13.4.11):
   A) Encryption algorithm;
   B) Key length; and
   C) SK_ei shared key (see 5.13.4.4);
2) Integrity check the decrypted data using the integrity algorithm specified in the MGMT_DATA SA parameter as follows:
   A) If the integrity algorithm is not AUTH_COMBINED, then compute an integrity check value for the decrypted data using the specified integrity algorithm and SK_ai shared key (see 5.13.4.4) stored in the MGMT_DATA SA parameter; or
   B) If the integrity algorithm is AUTH_COMBINED, then compute an integrity check value for the decrypted data in combination with decrypting the CIPHERTEXT field in step 1);
   and
3) Verify that the computed integrity check value matches the one contained in the INTEGRITY CHECK VALUE field in the Encrypted payload (see 7.6.3.5.10.1).

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. The IKEv2-SCSI CCS state being maintained for the I_T_L nexus on which the command was received, if any, shall not be affected by this error.

7.6.3.5.10.5 Decrypting an Encrypted payload that is part of an IKEv2-SCSI CCS

Before performing any checks of data contained in the Encrypted payload received in an Authentication step (see 5.13.4.9.2) or Delete operation (see 5.13.4.13) SECURITY PROTOCOL OUT parameter list, the device server shall decrypt and check the integrity of the Encrypted payload as follows:

1) Decrypt the CIPHERTEXT field of the Encrypted payload using the contents of the INITIALIZATION VECTOR field in the Encrypted payload (see 7.6.3.5.10.1), the AAD described in 7.6.3.5.10.2, and following information from the MGMT_DATA_SA parameter (see 5.13.4.11):
   A) Encryption algorithm;
   B) Key length; and
   C) SK_ei shared key (see 5.13.4.4);
2) Integrity check the decrypted data using the integrity algorithm specified in the MGMT_DATA SA parameter as follows:
   A) If the integrity algorithm is not AUTH_COMBINED, then compute an integrity check value for the decrypted data using the specified integrity algorithm and SK_ai shared key (see 5.13.4.4) stored in the MGMT_DATA SA parameter; or
   B) If the integrity algorithm is AUTH_COMBINED, then compute an integrity check value for the decrypted data in combination with decrypting the CIPHERTEXT field in step 1);
   and
3) Verify that the computed integrity check value matches the one contained in the INTEGRITY CHECK VALUE field in the Encrypted payload (see 7.6.3.5.10.1).

If the integrity checking of the Encrypted payload fails, then the error shall be processed as described in 7.6.3.8.2.

7.6.3.5.10.6 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 5.13.4.9.3), the application client should decrypt and check the integrity of the Encrypted payload as follows:

1) Decrypt the CIPHERTEXT field of the Encrypted payload using the contents of the INITIALIZATION VECTOR field in the Encrypted payload (see 7.6.3.5.10.1), the AAD described in 7.6.3.5.10.2, and following information from the MGMT_DATA_SA parameter (see 5.13.4.11):
   A) Encryption algorithm;
   B) Key length; and
   C) SK_er shared key (see 5.13.4.4);
2) Integrity check the decrypted data using the integrity algorithm specified in the MGMT_DATA SA parameter as follows:
   A) If the integrity algorithm is not AUTH_COMBINED, then compute an integrity check value for the decrypted data using the specified integrity algorithm and SK_ar shared key (see 5.13.4.4) stored in the MGMT_DATA SA parameter; or
   B) If the integrity algorithm is AUTH_COMBINED, then compute an integrity check value for the decrypted data in combination with decrypting the CIPHERTEXT field in step 1);
   and

Verify that the computed integrity check value matches the one contained in the INTEGRITY CHECK VALUE field in the Encrypted payload (see 7.6.3.5.10.1).

If the integrity checking of the Encrypted payload fails, then the application client should abandon the IKEv2-SCSI CCS and notify the device server that it is abandoning the IKEv2-SCSI CCS as described in 5.13.4.12.