ENDL
TEXAS

Date: 17 January 2008
To: T10 Technical Committee
From: Ralph O. Weber
Subject: ESP-SCSI for Parameter Data

This proposal defines a mechanism for applying SA parameter data to descriptors that are transferred in data-in buffer and/or data-out buffer parameter data.

Revision History

- r0 Original revision
- r1 Revise based on comments from April CAP Security WG, Matt Ball, and Paul Entzel
  Also increased the sequence count fields to 8 bytes (64 bits) and updated for combined encryption/integrity modes
- r2 Revised based on comments received from David Black, with reference to comments by Matt Ball.
- r3 Additional corrections made based on 29 October Security Conference Call review.
- r4 Additional corrections made to correctly compute integrity check values based on comments from Bob Nixon. All 06-449 references updated to match 07-437r4 (the approved revision of the SA creation doc). The descriptions for ENCRYPTED OR AUTHENTICATED DATA field and INTEGRITY CHECK VALUE field were modified significantly to accommodate combined mode encryption algorithms. Four occurrences of ESC-SCSI were corrected to ESP-SCSI.
- r5 As per the 18 December conference call, byte offset values were updated in all tables containing initialization vector fields.
- r6 Updated to match ongoing SA creation updates in SPC-4 r12 and 08-037.
- r7 Updated as requested by the January CAP WG.

Change bars indicate the differences between r6 and r7.

Related Documents

SPC-4 r12
T10/06-225r5 (Matt Ball, Quantum Corp.) "Using NIST AES Key-Wrap for Key Establishment"
T10/08-037r1 (Ralph Weber) "SA Creation corrections and clarifications"
IETF RFC 4303 "IP Encapsulating Security Payload (ESP)"
IETF RFC 4306 "Internet Key Exchange (IKEv2) Protocol"

Overview

The purpose of this proposal is to provide a way to transfer encrypted and/or integrity checked parameter data in data-in buffers and/or data-out buffers using IETF's Encapsulating Security Payload (ESP) which is specified in RFC 4303. The version described here is called ESP-SCSI, and is slightly different than IETF's version of ESP to provide consistency with the proposed usage in SCSI parameter data instead of the frame-oriented basis that underlies the ESP design.

To comply with FIPS 140-2, it is necessary to enter a key into a cryptographic module (i.e., a SCSI device server) using an approved encryption algorithm. ESP-SCSI is one method for satisfying this requirement.

Proposed SPC-4 Changes

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

{Note: Additions are shown in blue, deletions are shown in ~~red strikeout~~, and notes are shown in green.}

Introduction

Annex C identifies the differences between the various security standards upon which features in this standard are based (e.g., IKEv2 as defined in ~~(see~~ RFC 4306) and the ~~IKEv2 SCSI SA creation protocol~~ equivalent security features defined by this standard. (informative)

2.5 IETF References

Copies of the following approved IETF standards may be obtained through the Internet Engineering Task Force (IETF) at www.ietf.org.

…

RFC 4303, *IP Encapsulating Security Payload (ESP)*

3.1 Definitions

…

3.1.a Additional Authentication Data (AAD): A required input to encryption algorithms that also provide integrity checking.

3.1.e Encapsulating Security Payload for SCSI (ESP-SCSI): A method for transferring encrypted and/or integrity checked parameter data in data-in buffers and/or data-out buffers based on Encapsulating Security Payload (see RFC 4303). See 7.6.x.

3.1.i integrity check value: A value used to cryptographically validate the integrity of a specified set of bytes that contain specified data.

…

3.2 Symbols and acronyms

…

AAD	Additional Authentication Data ~~(see 7.7.3.5.10)~~ (see 3.1.a)
ESP-SCSI	Encapsulating Security Payload for SCSI (see 3.1.e)

5.13 Security Features

…

5.13.2.2 SA parameters

…

Table 45 — USAGE_TYPE SA parameter values

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

[a] The ESP-SCSI usage model is defined in 7.6.x

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

…

7.6.3.6.4 Integrity algorithm (INTEG) IKEv2-SCSI cryptographic algorithm descriptor

…

The ALGORITHM IDENTIFIER field (see table x388) specifies integrity checking algorithm and shared key length to which the INTEG IKEv2-SCSI cryptographic algorithm descriptor applies.

Table 388 — INTEG ALGORITHM IDENTIFIER field

Code	IKEv2 Name	ICV [a] length (bytes)	Key length (bytes)	Support	Reference
8003 0002h	AUTH_HMAC_SHA1_96	12	20	Optional	RFC 2404
8003 000Ch	AUTH_HMAC_SHA2_256_128	16	32	Optional	RFC 4868
8003 000Eh	AUTH_HMAC_SHA2_512_256	32	64	Optional	RFC 4868
F003 0001h	AUTH_COMBINED	0	0	Optional	this subclause
8003 0400h – 8003 FFFFh	Vendor Specific				
0000 0000h – 0000 FFFFh	Restricted				IANA
All others	Reserved				

[a] Integrity Check Value.

{Note: The remainder of this proposal is new text, but only the first new subclause header is shown in blue.}

7.6.x ESP-SCSI for parameter data

7.6.x.1 Overview

Subclause 7.6.x defines a method for transferring encrypted and/or integrity checked parameter data in data-in buffers and/or data-out buffers. The method is based on the Encapsulating Security Payload (see RFC 4303) standard developed by the IETF. Because of the constrained usage of ESP-SCSI parameter data in data-in buffers and/or data-out buffers, the method defined in this standard differs from the one found in RFC 4303.

7.6.x.2 ESP-SCSI required inputs

Prior to using the ESP-SCSI descriptors defined in 7.6.x, an SA shall be created (see 5.13.2.3) with SA parameters (see 5.13.2.2) that conform to the requirements defined in 5.13.2.3 and to the following:

a) The USAGE_TYPE SA parameter shall be set to a value for which ESP-SCSI usage is defined (see table 45);
b) The USAGE_DATA SA parameter shall contain at least the following:
   A) The algorithm identifier and key length for the encryption algorithm (e.g., the ALGORITHM IDENTIFIER field and KEY LENGTH field from the ENCR IKEv2-SCSI cryptographic algorithm descriptor (see 7.6.3.6.2) in the IKEv2-SCSI SAUT Cryptographic Algorithms payload (see 7.6.3.5.13) negotiated by an IKEv2-SCSI SA creation protocol (see 5.13.4)); and
   B) The algorithm identifier for the integrity algorithm (e.g., the ALGORITHM IDENTIFIER field from the INTEG IKEv2-SCSI cryptographic algorithm descriptor (see 7.6.3.6.4)in the IKEv2-SCSI SAUT Cryptographic Algorithms payload (see 7.6.3.5.13) negotiated by an IKEv2-SCSI SA creation protocol (see 5.13.4));
   and
c) The KEYMAT SA parameter shall consist of the shared keys described in 5.13.4.10.6.

ESP-SCSI depends on the following additional information derived from the contents of the USAGE_DATA SA parameter:

a) The encryption algorithm identifier shall indicate:
   A) The absence of encryption by having the value ENCR_NULL (see table 59 in 5.13.7);
   B) The size of the initialization vector, if any (e.g., as shown in table 384 (see 7.6.3.6.2));
   C) The size of the salt bytes, if any (e.g., as shown in table 384 (see 7.6.3.6.2));
   D) For combined mode encryption algorithms, the size of the integrity check value (i.e., the algorithm's MAC size as shown in table 384 (see 7.6.3.6.2));
   and
b) The integrity algorithm identifier shall indicate:
   A) The use of a combined mode encryption algorithm by having the value AUTH_COMBINED (see table 59 in 5.13.7);
   B) For non-combined mode encryption algorithms, the size of the integrity check value (see table 388 in 7.6.3.6.4).

Each shared key in KEYMAT shall be taken from the KDF generated bits in the order shown in 5.13.4.10.6. The size of each of the shared keys in KEYMAT is determined by the negotiated encryption algorithm and integrity algorithm as described in 5.13.4.4.

7.6.x.3 ESP-SCSI data format before encryption and after decryption

Before data bytes are encrypted and after they are decrypted, they have the format shown in table x1.

Table x1 — ESP-SCSI data format before encryption and after decryption

Bit / Byte	7	6	5	4	3	2	1	0
0	UNENCRYPTED BYTES							
p-1								
p	PADDING BYTES							
j-1								
j	PAD LENGTH (j-p)							
j+1	MUST BE ZERO							

The UNENCRYPTED BYTES field contains the bytes that are to be protected via encryption or that have been decrypted.

Before encryption, the PADDING BYTES field contains zero to 255 bytes. The number of padding bytes is:

a) Defined by the encryption algorithm; or
b) The number needed to cause the length of all bytes prior to encryption (i.e., j+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 command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, the additional sense code set to INVALID FIELD IN PARAMETER LIST, the SKSV bit set to one, and SENSE KEY SPECIFIC field set to indicate the last byte in the encrypted data as defined in 4.5.2.4.2. 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.

The MUST BE ZERO field contains zero. After decryption, the contents of the MUST BE ZERO field shall be verified to be zero. If this verification is not successful in a device server, the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, the additional sense code set to INVALID FIELD IN PARAMETER LIST, the SKSV bit set to one, and SENSE KEY SPECIFIC field set to indicate the last byte in the encrypted data as defined in 4.5.2.4.2. If this verification is not successful in an application client, the decrypted data should be ignored.

7.6.x.4 ESP-SCSI data-out buffer parameter list data descriptors

7.6.x.4.1 Overview

When ESP-SCSI is used in parameter list data that appears in a data-out buffer, the parameter list data contains one or more descriptors selected based on the criteria shown in table x2.

Table x2 — ESP-SCSI data-out buffer parameter data descriptors

Descriptor name	External descriptor length [a]	Initialization vector present [b]	Reference
ESP-SCSI out	No	No	table x3 in 7.6.x.4.2
	No	Yes	table x4 in 7.6.x.4.2
ESP-SCSI out w/o length	Yes	No	table x5 in 7.6.x.4.3
	Yes	Yes	table x6 in 7.6.x.4.3

[a] This is determined by the data format defined for the data-out buffer parameter data. If the format includes a length for the ESP-SCSI descriptor, then the answer to this question is yes.

[b] This is determined from the USAGE_DATA SA parameter (see 7.6.x.2).

7.6.x.4.2 ESP-SCSI data-out buffer parameter lists including a descriptor length

If the USAGE_DATA SA parameter (see 7.6.x.2) indicates an encryption algorithm whose initialization vector size is zero, then the data-out buffer parameter list descriptor shown in table x3 contains the ESP-SCSI data.

Table x3 — ESP-SCSI data-out buffer parameter list descriptor without initialization vector

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)							
1				DESCRIPTOR LENGTH (n-1)				(LSB)
2								
3				Reserved				
4	(MSB)							
7				DS_SAI				(LSB)
8	(MSB)							
15				DS_SQN				(LSB)
16								
i-1				ENCRYPTED OR AUTHENTICATED DATA				
i	(MSB)							
n				INTEGRITY CHECK VALUE				(LSB)

The DESCRIPTOR LENGTH field, DS_SAI field, DS_SQN field, ENCRYPTED OR AUTHENTICATED DATA field, and INTEGRITY CHECK VALUE field are defined after table x4 in this subclause.

If the USAGE_DATA SA parameter indicates an indicates an encryption algorithm whose initialization vector size (i.e., s) is greater than zero, the data-out buffer parameter data descriptor shown in table x4 contains the ESP-SCSI data.

Table x4 — ESP-SCSI data-out buffer full parameter list descriptor

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)							
1				DESCRIPTOR LENGTH (n-1)				(LSB)
2								
3				Reserved				
4	(MSB)							
7				DS_SAI				(LSB)
8	(MSB)							
15				DS_SQN				(LSB)
16	(MSB)							
16+s-1				INITIALIZATION VECTOR				(LSB)
16+s								
i-1				ENCRYPTED OR AUTHENTICATED DATA				
i	(MSB)							
n				INTEGRITY CHECK VALUE				(LSB)

The DESCRIPTOR LENGTH field specifies the number of bytes that follow in the ESP-SCSI data-out buffer parameter list descriptor.

The DS_SAI field contains the value in the DS_SAI SA parameter (see 5.13.2.2) for the SA that is being used to prepare the ESP-SCSI data-out buffer parameter list descriptor. If the DS_SAI value is not known to the device server, the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, the additional sense code set to INVALID FIELD IN PARAMETER LIST, the SKSV bit set to one, and SENSE KEY SPECIFIC field set as defined in 4.5.2.4.2.

The DS_SQN field should contain one plus the value in the application client's DS_SQN SA parameter (see 5.13.2.2) for the SA that is being used to prepare the ESP-SCSI data-out buffer parameter list descriptor. Before sending the ESP-SCSI data-out buffer parameter list, the application client should copy the contents of the DS_SQN field to its DS_SQN SA parameter.

The device server shall terminate the command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, the additional sense code set to INVALID FIELD IN PARAMETER LIST, the SKSV bit set to one, and SENSE KEY SPECIFIC field set as defined in 4.5.2.4.2 if any of the following conditions are detected:

a) The DS_SQN field is set to zero;
b) The value in the DS_SQN field is less than or equal to the value in the device server's DS_SQN SA parameter; or
c) The value in the DS_SQN field is greater than 32 plus the value in the device server's DS_SQN SA parameter.

If the DS_SQN SA parameter is equal to FFFF FFFF FFFF FFFFh, the device server shall delete the SA.

The INITIALIZATION VECTOR field, if any, contains a value that is used as an input into the encryption algorithm and/or integrity algorithm specified by the SA specified by the DS_SAI field. The INITIALIZATION VECTOR field is not encrypted. The encryption algorithm and/or integrity algorithm may define additional requirements for the INITIALIZATION VECTOR field.

The ENCRYPTED OR AUTHENTICATED DATA field contains:

a) If an encryption algorithm for the SA specified by the DS_SAI field is not ENCR_NULL, encrypted data bytes for the following:
   1) The bytes in the UNENCRYPTED BYTES field (see 7.6.x.3);
   2) The bytes in the PADDING BYTES field (see 7.6.x.3);
   3) The PAD LENGTH field byte (see 7.6.x.3); and
   4) The MUST BE ZERO field byte (see 7.6.x.3);

   or
b) Otherwise, the unencrypted data bytes.

If the integrity algorithm for the SA specified by the DS_SAI field is AUTH_COMBINED (see 7.6.x.2), then the AAD input to the encryption algorithm is composed of the following bytes, in order:

1) The bytes in the DS_SAI field; and
2) The bytes in the DS_SQN field;

The INTEGRITY CHECK VALUE field contains a value that is computed as follows:

a) If the integrity algorithm is not AUTH_COMBINED, the integrity check value is computed using the specified integrity algorithm with the following bytes as inputs, in order:
   1) The bytes in the DS_SAI field;
   2) The bytes in the DS_SQN field;
   3) The bytes in the INITIALIZATION VECTOR field, if any; and
   4) The bytes in the ENCRYPTED OR AUTHENTICATED DATA field after encryption, if any, has been performed;

   or
b) If the integrity algorithm is AUTH_COMBINED, the integrity check value is computed as an additional output of the specified encryption algorithm.

Upon receipt of ESP-SCSI data-out buffer parameter data, the device server shall compute an integrity check value for the ESP-SCSI parameter data as specified by the algorithms specified by the SA specified by the DS_SAI field using the inputs shown in this subclause. If the computed integrity check value does not match the value in the INTEGRITY CHECK VALUE field, the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, the additional sense code set to INVALID FIELD IN PARAMETER LIST, the SKSV bit set to one, and SENSE KEY SPECIFIC field set as defined in 4.5.2.4.2.

If the command is not terminated due to a sequence number error or a mismatch between the computed integrity check value and the contents of the INTEGRITY CHECK VALUE field, then the device server shall copy the contents of the received DS_SQN field to its DS_SQN SA parameter.

7.6.x.4.3 ESP-SCSI data-out buffer parameter lists for externally specified descriptor length

If the USAGE_DATA SA parameter (see 5.13.2.2) indicates an encryption algorithm whose initialization vector size is zero and the length of the ESP-SCSI data-out buffer parameter list descriptor appears elsewhere in the parameter list, then the data-out buffer parameter list descriptor shown in table x5 contains the ESP-SCSI data.

Table x5 — ESP-SCSI data-out buffer parameter list descriptor without length and initialization vector

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)			DS_SAI				
3								(LSB)
4	(MSB)			DS_SQN				
11								(LSB)
12				ENCRYPTED OR AUTHENTICATED DATA				
i-1								
i	(MSB)			INTEGRITY CHECK VALUE				
n								(LSB)

The DS_SAI field, DS_SQN field, ENCRYPTED OR AUTHENTICATED DATA field, and INTEGRITY CHECK VALUE field are defined in 7.6.x.4.2.

If the USAGE_DATA SA parameter indicates an indicates an encryption algorithm whose initialization vector size (i.e., s) is greater than zero and the length of the ESP-SCSI data-out buffer parameter list descriptor appears elsewhere in the parameter list, the data-out buffer parameter list descriptor shown in table x6 contains the ESP-SCSI data.

Table x6 — ESP-SCSI data-out buffer parameter list descriptor without length

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)			DS_SAI				
3								(LSB)
4	(MSB)			DS_SQN				
11								(LSB)
12	(MSB)			INITIALIZATION VECTOR				
12+s-1								(LSB)
12+s				ENCRYPTED OR AUTHENTICATED DATA				
i-1								
i	(MSB)			INTEGRITY CHECK VALUE				
n								(LSB)

The DS_SAI field, DS_SQN field, INITIALIZATION VECTOR field, ENCRYPTED OR AUTHENTICATED DATA field, and INTEGRITY CHECK VALUE field are defined in 7.6.x.4.2.

7.6.x.5 ESP-SCSI data-in buffer parameter data descriptors

7.6.x.5.1 Overview

A device server shall transfer ESP-SCSI parameter data descriptors in a data-in buffer only in response to a request that specifies an SA using the AC_SAI SA parameter and DS_SAI SA parameter values (see 5.13.2.2). If the specified combination of AC_SAI and DS_SAI values in a command that requests the transfer of ESP-SCSI parameter data descriptors is not known to the device server, the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, the additional sense code set to INVALID FIELD IN PARAMETER LIST or to INVALID FIELD IN CDB, the SKSV bit set to one, and SENSE KEY SPECIFIC field set as defined in 4.5.2.4.2.

When ESP-SCSI is used in parameter data which appears in a data-in buffer, the parameter data contains one or more descriptors selected based on the criteria shown in table x7.

Table x7 — ESP-SCSI data-in buffer parameter data descriptors

Descriptor name	External descriptor length [a]	Initialization vector present [b]	Reference
ESP-SCSI in	No	No	table x8 in 7.6.x.5.2
	No	Yes	table x9 in 7.6.x.5.2
ESP-SCSI in w/o length	Yes	No	table x10 in 7.6.x.5.3
	Yes	Yes	table x11 in 7.6.x.5.3

[a] This is determined by the data format defined for the data-in buffer parameter data. If the format includes a length for the ESP-SCSI descriptor, then the answer to this question is yes.

[b] This is determined from the USAGE_DATA SA parameter (see 7.6.x.2).

7.6.x.5.2 ESP-SCSI data-in buffer parameter data including a descriptor length

If the USAGE_DATA SA parameter (see 5.13.2.2) indicates an encryption algorithm whose initialization vector size is zero, then the data-in buffer parameter data descriptor shown in table x8 contains the ESP-SCSI data.

Table x8 — ESP-SCSI data-in buffer parameter data descriptor without initialization vector

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)							
1				DESCRIPTOR LENGTH (n-1)				(LSB)
2								
3				Reserved				
4	(MSB)							
7				AC_SAI				(LSB)
8	(MSB)							
15				AC_SQN				(LSB)
16								
i-1				ENCRYPTED OR AUTHENTICATED DATA				
i	(MSB)							
n				INTEGRITY CHECK VALUE				(LSB)

The DESCRIPTOR LENGTH field, AC_SAI field, AC_SQN field, ENCRYPTED OR AUTHENTICATED DATA field, and INTEGRITY CHECK VALUE field are defined after table x9 in this subclause.

If the USAGE_DATA SA parameter indicates an indicates an encryption algorithm whose initialization vector size (i.e., s) is greater than zero, the data-in buffer parameter data descriptor shown in table x9 contains the ESP-SCSI data.

Table x9 — ESP-SCSI data-in buffer full parameter data descriptor

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)							
1				DESCRIPTOR LENGTH (n-1)				(LSB)
2								
3				Reserved				
4	(MSB)							
7				AC_SAI				(LSB)
8	(MSB)							
15				AC_SQN				(LSB)
16	(MSB)							
16+s-1				INITIALIZATION VECTOR				(LSB)
16+s								
i-1				ENCRYPTED OR AUTHENTICATED DATA				
i	(MSB)							
n				INTEGRITY CHECK VALUE				(LSB)

The DESCRIPTOR LENGTH field specifies the number of bytes that follow in the ESP-SCSI data-in buffer parameter data descriptor.

The AC_SAI field contains the value in the AC_SAI SA parameter (see 5.13.2.2) for the SA that is being used to prepare the ESP-SCSI data-in buffer parameter data descriptor. If the AC_SAI value is not known to the application client, the ESP-SCSI data-in parameter data descriptor should be ignored.

The AC_SQN field contains one plus the value in the device server's AC_SQN SA parameter (see 5.13.2.2) for the SA that is being used to prepare the ESP-SCSI data-on buffer parameter data descriptor. Before sending the ESP-SCSI data-out buffer parameter list as part of a command that completes with GOOD status, the device server shall copy the contents of the AC_SQN field to its AC_SQN SA parameter. The device server shall not send two ESP-SCSI data-out buffer parameter data descriptors that contain the same values in AC_SAI field and AC_SQN field.

If the AC_SQN SA parameter is equal to FFFF FFFF FFFF FFFFh, the device server shall delete the SA after the data-in buffer parameter data containing that value is sent.

The application client should ignore the ESP-SCSI data-in parameter data descriptor if any of the following conditions are detected:

a) The AC_SQN field is set to zero;
b) The value in the AC_SQN field is less than or equal to the value in the application client's AC_SQN SA parameter; or
c) The value in the AC_SQN field is greater than 32 plus the value in the application client's AC_SQN SA parameter.

The INITIALIZATION VECTOR field, if any, contains a value that is used as an input into the encryption algorithm and/or integrity algorithm specified by the SA specified by the AC_SAI field. The INITIALIZATION VECTOR field is not encrypted. The encryption algorithm and/or integrity algorithm may define additional requirements for the INITIALIZATION VECTOR field.

The ENCRYPTED OR AUTHENTICATED DATA field contains:

a) If an encryption algorithm specified by the SA specified by the AC_SAI field is not ENCR_NULL, encrypted data bytes for the following:
   1) The bytes in the UNENCRYPTED BYTES field (see 7.6.x.3);
   2) The bytes in the PADDING BYTES field (see 7.6.x.3);
   3) The PAD LENGTH field byte (see 7.6.x.3); and
   4) The MUST BE ZERO field byte (see 7.6.x.3);

   or
b) Otherwise, the unencrypted data bytes.

If the integrity algorithm for the SA specified by the AC_SAI field is AUTH_COMBINED (see 7.6.x.2), then the AAD input to the encryption algorithm is composed of the following bytes, in order:

1) The bytes in the AC_SAI field; and
2) The bytes in the AC_SQN field;

The INTEGRITY CHECK VALUE field contains a value that is computed as follows:

a) If the integrity algorithm is not AUTH_COMBINED, the integrity check value is computed using the specified integrity algorithm with the following bytes as inputs, in order:
   1) The bytes in the AC_SAI field;
   2) The bytes in the AC_SQN field;
   3) The bytes in the INITIALIZATION VECTOR field, if any; and
   4) The bytes in the ENCRYPTED OR AUTHENTICATED DATA field after encryption, if any, has been performed;

   or
b) If the integrity algorithms is AUTH_COMBINED, the integrity check value is computed as an additional output of the specified encryption algorithm.

Upon receipt of ESP-SCSI data-in buffer parameter data, the application client should compute an integrity check value for the ESP-SCSI parameter data as specified by the algorithms specified by the SA specified by the AC_SAI field using the inputs shown in this subclause. If the computed integrity check value does not match the value in the INTEGRITY CHECK VALUE field, the results returned by the command should be ignored.

The application client should copy the contents of the AC_SQN field to its AC_SQN SA parameter if all of the following are true:

a) The command completed with GOOD status;
b) The ESP-SCSI data-in parameter data descriptor was not ignored due to inconsistency problems with the AC_SQN field; and
c) The computed integrity check value matched the contents of the INTEGRITY CHECK VALUE field.

7.6.x.5.3 ESP-SCSI data-in buffer parameter data for externally specified descriptor length

If the USAGE_DATA SA parameter (see 5.13.2.2) indicates an encryption algorithm whose initialization vector size is zero and the length of the ESP-SCSI data-in buffer parameter data descriptor appears elsewhere in the parameter data, then the data-in buffer parameter data descriptor shown in table x10 contains the ESP-SCSI data.

Table x10 — ESP-SCSI data-in buffer parameter data descriptor without length and initialization vector

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)			AC_SAI				
3								(LSB)
4	(MSB)			AC_SQN				
11								(LSB)
12				ENCRYPTED OR AUTHENTICATED DATA				
i-1								
i	(MSB)			INTEGRITY CHECK VALUE				
n								(LSB)

The AC_SAI field, AC_SQN field, ENCRYPTED OR AUTHENTICATED DATA field, and INTEGRITY CHECK VALUE field are defined in 7.6.x.5.2.

If the USAGE_DATA SA parameter indicates an indicates an encryption algorithm whose initialization vector size (i.e., s) is greater than zero and the length of the ESP-SCSI data-in buffer parameter data descriptor appears elsewhere in the parameter data, the data-in buffer parameter data descriptor shown in table x11 contains the ESP-SCSI data.

Table x11 — ESP-SCSI data-in buffer parameter data descriptor without length

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)			AC_SAI				
3								(LSB)
4	(MSB)			AC_SQN				
11								(LSB)
12	(MSB)			INITIALIZATION VECTOR				
12+s-1								(LSB)
12+s				ENCRYPTED OR AUTHENTICATED DATA				
i-1								
i	(MSB)			INTEGRITY CHECK VALUE				
n								(LSB)

The AC_SAI field, AC_SQN field, INITIALIZATION VECTOR field, ENCRYPTED OR AUTHENTICATED DATA field, and INTEGRITY CHECK VALUE field are defined in 7.6.x.5.2.

{{Modify Annex C as shown here.}}

Annex C

(Informative)

~~IKEv2 protocol details and variations for IKEv2-SCSI~~

Notes regarding security features

C.1 IKEv2 protocol details and variations for IKEv2-SCSI

{{No other changes.}}

C.2 ESP protocol details and variations for ESP-SCSI

{{All of C.2 is new. Changes markups suspended.}}

The IKEv2 protocol details and variations specified in RFC 4303 apply to ESP-SCSI (i.e., this standard) as follows:

a) This standard requires an integrity check value (ICV field), whereas ESP allows support of confidentiality-only;
b) This standard does not support traffic flow confidentiality;
c) This standard does not support the TCP/IP aspects of ESP (e.g., IP addresses, multicast);
d) This standard requires anti-replay detection using the sequence number, whereas ESP makes this optional;
e) This standard does not support the Next Header field, but does reserve space for it in the MUST BE ZERO field (see table x1 in 7.6.x.3);
f) This standard requires verification of the padding bytes, when possible;
g) There is no provision in this standard for generating 'dummy packets'; and
h) This standard does not support out-of-order parameter data.