ENDL
TEXAS

Date: 10 April 2007
To: T10 Technical Committee
From: Ralph O. Weber w/ help from Matt Ball & David Black
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

Related Documents

T10/06-225r5 (Matt Ball, Quantum Corp.) "Using NIST AES Key-Wrap for Key Establishment"
T10/06-369r8 (Ralph Weber, ENDL Texas) "Security Association Model for SPC-4"
T10/06-449r2 (Matt Ball and David Black) "SPC-4: Establishing a Security Association using IKEv2"
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 data-origin authenticated 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

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

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.e Encapsulating Security Payload for SCSI (ESP-SCSI): a method for transferring encrypted and/or data origin authenticated parameter data in data-in buffers and/or data-out buffers based on Encapsulating Security Payload (see RFC 4303). See 5.13.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

…

ESP-SCSI	Encapsulating Security Payload for SCSI (see 3.1.e)

5.13 Security Features

…

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

5.13.x ESP-SCSI for parameter data

5.13.x.1 Overview

Subclause 5.13.x defines a method for transferring encrypted and/or data origin authenticated 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 in 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.

5.13.x.2 ESP-SCSI required inputs

Prior to using the ESP-SCSI descriptors defined in 5.13.x, an SA shall be created (see 5.13.a, defined in 06-449) with SA parameters (see 5.13.2.2, defined in 06-369r8) that conform to the requirements defined in 5.13.2.3 (see 06-369r8) and to the following:

a) The USAGE_TYPE SA parameter shall be set to a value for which ESP-SCSI usage is defined;
b) The USAGE_DATA SA parameter shall contain at least the following:
    a) The size in bytes of the initialization vector for the negotiated encryption algorithm; and
    b) The size in bytes of the integrity check value for the:
        a) If the negotiated encryption algorithm includes an integrity check value, the size for the negotiated encryption algorithm; or
        b) The size for the negotiated integrity algorithm;

    and
c) The KEYMAT SA parameter shall consist of the following:
    1) The shared key used by the integrity algorithm that is applied to data-out buffers (i.e., SK_ai);
    2) The shared key used by the integrity algorithm that is applied to data-in buffers (i.e., SK_ar);
    3) The shared key used by the encryption algorithm that is applied to data-out buffers (i.e., SK_ei); and
    4) The shared key used by the encryption algorithm that is applied to data-in buffers (i.e., SK_er).

Each shared key in KEYMAT shall be taken, in order, from the KDF generated bits (see 5.13.3, defined in 06-369r8). The size of each of the shared keys in KEYMAT is determined by the negotiated encryption algorithm and integrity algorithm. If there is no integrity algorithm (e.g., integrity algorithm is IKEv2-SCSI NONE), then the sizes of SK_ai and SK_ar are zero.

Note n1: Some algorithms, such as galois/counter mode (i.e., GCM) and counter mode encryption with cipher block chaining message authentication code (i.e., CCM), use part of the shared key for a nonce that it implicitly includes as part of the initialization vector. The encryption algorithm describes the specific usage of the shared key.

5.13.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							
l-1								
l	PAD LENGTH (l-p)							
l+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 minimum number needed to cause the length of all bytes prior to encryption (i.e., l+2) to be a whole multiple of the cipher block size for the encryption algorithm being used.

{Devotees of ESP are profoundly offended by the word 'minimum' in b) above.}

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 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 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.

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

5.13.x.4.1 Overview

When ESP-SCSI is used in parameter list data which 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	Initialization vector	Reference
ESP-SCSI out	No	No	table x3 in 5.13.x.4.2
	No	Yes	table x4 in 5.13.x.4.2
ESP-SCSI out w/o length	Yes	No	table x5 in 5.13.x.4.3
	Yes	Yes	table x6 in 5.13.x.4.3

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

If the USAGE_DATA SA parameter (see 5.13.2.2, defined in 06-369r8) indicates an initialization vector size of 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)	DESCRIPTOR LENGTH (n-1)						
1								(LSB)
2	(MSB)	DS_SAI						
5								(LSB)
6	(MSB)	DS_SQN						
9								(LSB)
10		ENCRYPTED OR AUTHENTICATED DATA						
i-1								
i	(MSB)	INTEGRITY CHECK VALUE						
n								(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 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)	DESCRIPTOR LENGTH (n-1)						
1								(LSB)
2	(MSB)	DS_SAI						
5								(LSB)
6	(MSB)	DS_SQN						
9								(LSB)
10	(MSB)	INITIALIZATION VECTOR						
10+s-1								(LSB)
10+s		ENCRYPTED OR AUTHENTICATED DATA						
i-1								
i	(MSB)	INTEGRITY CHECK VALUE						
n								(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, defined in 06-369r8) 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 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 the least-significant four bytes from value in the DS_SQN SA parameter (see 5.13.2.2, defined in 06-369r8) for the SA that is being used to prepare the ESP-SCSI data-out buffer parameter list descriptor. 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;
c) The value in the DS_SQN field is greater than 32 plus the value in the device server's DS_SQN SA parameter; or
d) The value in the device server's DS_SQN SA parameter is greater than 2^{32} minus one.

If the command is not terminated due to sequence number errors:

a) The device server shall add one to the value in its DS_SQN SA parameter; and
b) The application client should add one to the value in its DS_SQN SA parameter.

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

The ENCRYPTED OR AUTHENTICATED DATA field contains:

a) If an encryption algorithm is specified by the SA specified by the DS_SAI field, encrypted data bytes; or
b) unencrypted data bytes in all other cases.

Unless otherwise specified by the encryption algorithm or authentication algorithm, the INTEGRITY CHECK VALUE field contains a value that is computed as specified by the algorithms specified by the SA specified by the DS_SAI field. The integrity check value is computed using 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; and
4) The following bytes based on whether encryption is being performed:
    a) If an encryption algorithm is specified by the SA specified by the DS_SAI field:
        1) The bytes in the UNENCRYPTED BYTES field (see 5.13.x.3);
        2) The bytes in the PADDING BYTES field (see 5.13.x.3);
        3) The PAD LENGTH field byte (see 5.13.x.3); and
        4) The MUST BE ZERO field byte (see 5.13.x.3);
        or
    b) If an encryption algorithm is not specified by the SA specified by the DS_SAI field:
        1) The bytes in the ENCRYPTED OR AUTHENTICATED DATA field.

5.13.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, defined in 06-369r8) indicates an initialization vector size of zero and the length of the ESC-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				
7								(LSB)
8				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 5.13.x.4.2.

If the USAGE_DATA SA parameter indicates an indicates an initialization vector size (i.e., s) is greater than zero and the length of the ESC-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				
7								(LSB)
8	(MSB)			INITIALIZATION VECTOR				
8+s-1								(LSB)
8+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 5.13.x.4.2.

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

5.13.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 DC_SAI SA parameter values (see 5.13.2.2, defined in 06-369r8). If the specified combination of AC_SAI and DC_SAI values in a command that requests the transfer of ESP-SCSI parameter data descriptors is not known to the device server, 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 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	Initialization Vector	Reference
ESP-SCSI in	No	No	table x8 in 5.13.x.5.2
	No	Yes	table x9 in 5.13.x.5.2
ESP-SCSI in w/o length	Yes	No	table x10 in 5.13.x.5.3
	Yes	Yes	table x11 in 5.13.x.5.3

5.13.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, defined in 06-369r8) indicates an initialization vector size of 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)			DESCRIPTOR LENGTH (n-1)				
1								(LSB)
2	(MSB)			AC_SAI				
5								(LSB)
6	(MSB)			AC_SQN				
9								(LSB)
10				ENCRYPTED OR AUTHENTICATED DATA				
i-1								
i	(MSB)			INTEGRITY CHECK VALUE				
n								(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 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)			DESCRIPTOR LENGTH (n-1)				
1								(LSB)
2	(MSB)			AC_SAI				
5								(LSB)
6	(MSB)			AC_SQN				
9								(LSB)
10	(MSB)			INITIALIZATION VECTOR				
10+s-1								(LSB)
10+s				ENCRYPTED OR AUTHENTICATED DATA				
i-1								
i	(MSB)			INTEGRITY CHECK VALUE				
n								(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, defined in 06-369r8) 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 should contain the least-significant four bytes from value in the AC_SQN SA parameter (see 5.13.2.2, defined in 06-369r8) for the SA that is being used to prepare the ESP-SCSI data-on buffer parameter data descriptor. 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;
c) The value in the AC_SQN field is greater than 32 plus the value in the application client's AC_SQN SA parameter; or
d) The value in the application's AC_SQN SA parameter is greater than 2^{32} minus one.

If the command completes with GOOD status:

a) The device server shall add one to the value in its AC_SQN SA parameter; and
b) The application client should add one to the value in its AC_SQN SA parameter.

{Is the above right?}

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

The ENCRYPTED OR AUTHENTICATED DATA field contains:

a) If an encryption algorithm is specified by the SA specified by the AC_SAI field, encrypted data bytes; or
b) unencrypted data bytes in all other cases.

Unless otherwise specified by the encryption algorithm or authentication algorithm, the INTEGRITY CHECK VALUE field contains a value that is computed as specified by the algorithms specified by the SA specified by the AC_SAI field. The integrity check value is computed using 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; and
4) The following bytes based on whether encryption is being performed:
    a) If an encryption algorithm is specified by the SA specified by the AC_SAI field:
        1) The bytes in the UNENCRYPTED BYTES field (see 5.13.x.3);
        2) The bytes in the PADDING BYTES field (see 5.13.x.3);
        3) The PAD LENGTH field byte (see 5.13.x.3); and
        4) The MUST BE ZERO field byte (see 5.13.x.3);

        or
    b) If an encryption algorithm is not specified by the SA specified by the AC_SAI field:
        1) The bytes in the ENCRYPTED OR AUTHENTICATED DATA field.

5.13.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, defined in 06-369r8) indicates an initialization vector size of zero and the length of the ESC-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				
7								(LSB)
8				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 5.13.x.5.2.

If the USAGE_DATA SA parameter indicates an indicates an initialization vector size (i.e., s) is greater than zero and the length of the ESC-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				
7								(LSB)
8	(MSB)			INITIALIZATION VECTOR				
8+s-1								(LSB)
8+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 5.13.x.5.2.