To: INCITS T10 Committee
From: Matt Ball, Quantum Corporation
Date: March 13, 2007
Subject: SSC-3: Key Entry using Encapsulating Security Payload (ESP)

Revision History

Revision 0 (T10/06-225r0):
Posted to the T10 web site on 4 May 2006.

Revision 1 (T10/06-225r1) 18 May 2006:
Changed SPI to SAI.
Added definition for 'encrypted information packet'.
Removed 'Secure Channel'.
Added mention of ANS X9.102 as a standard that could replace AES Key Wrap.
Within **Table Y1 – Set Data Encryption page**, moved the SAI (formerly SPI) and SEQUENCE NUMBER fields into the KEY field and left those as 'reserved' (as they were previously). Table Y1 is now unchanged.
Updated **Table Y9 – key field contents with key format field set to 02h** to include SAI and SEQUENCE NUMBER fields.
Minor edits.

Revision 2 (T10/06-225r2) 6 June 2006:
Moved the security association parameters into a table.
Created additional level 5 subclauses for describing the various key formats.
Added changes to the references section
Added clauses for these topics: Nonces and Key derivation functions
Changed the layout for the 'key reference' format
Changed key-wrap reference to RFC 3394 instead of "AES Key-Wrap (Draft)"
Added table for assigning an ID to the key derivation functions (KDF).
Numerous rewordings.

Revision 3 (T10/06-225r3) 27 June 2006:
Edits based on feedback from Michael Banther

Revision 4 (T10/06-225r4) 2 Jan 2007:
Changed from using AES-Key to IETF's ESP

Revision 5, 11 March 2007:
Changes based on January meeting

Related Documents

T10/SSC-3 r3b "SCSI Stream Commands 3"
T10/06-225r3 (Matt Ball, Quantum Corp.) "Using NIST AES Key-Wrap for Key Establishment"
T10/06-369r6 (Ralph Weber, ENDL Texas) "Security Association Model for SPC-4"
T10/06-449r1 (Matt Ball and David Black) "SPC-4: Establishing a Security Association using IKEv2"
T11 FC-SP/06-157v2 "Fibre Channel Security Protocols (FC-SP)"
NIST FIPS 140-2 "Security Requirements for Cryptographic Modules"
NIST FIPS 140-2, Annex D "Approved Key Establishment Techniques"
NIST FIPS 140-2 IG "Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program"
NIST FIPS 180-2 “Secure Hash Standard (SHS)”
NIST FIPS 197 "Announcing the Advanced Encryption Standard (AES)"
NIST SP 800-38B "Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication"
NIST SP 800-56A "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"
NIST SP 800-57 "Recommendation on Key Management"
NIST SP 800-90 "Recommendation for Random Number Generation Using Deterministic Random Bit Generators"
IEEE 802.11i "Amendment 6: Medium Access Control (MAC) Security Enhancements"
IETF RFC 3394 "Advanced Encryption Standard (AES) Key Wrap Algorithm"

IETF RFC 4303 "IP Encapsulating Security Payload (ESP)"
IETF RFC 4306 “Internet Key Exchange (IKEv2) Protocol”

General

NOTE – This proposal assumes the incorporation of 06-369r6 and 06-449r1 into SPC-4.

The purpose of this proposal is to provide a way for the application client to pass an encrypted key to the device server (note: this is different from passing an *encryption* key; the goal is to encrypt the encryption key). This feature has some of the following benefits:

- It is possible to hide the encryption key from an eavesdropper. Generally, the encryption key is more valuable than the data, so it is important to keep this information safe.
- To comply with NIST FIPS 140-2, it is necessary to encrypt the key before passing it to the device server. Currently, there is no method in SSC-3 to provide a FIPS-compliant solution.

To comply with FIPS 140-2, it is necessary to enter a key into a cryptographic module (device server) using an Approved encryption algorithm. NIST has currently approved CCM, and will soon approve GCM.

This proposal borrows several concepts from the IETF, IEEE, and NIST. The message format is very similar to that of RFC 4303, "IP Encapsulating Security Payload (ESP)". The *key derivation function* (KDF) comes from a recent standard by NIST (SP 800-56A). The reader is encouraged to review these documents and other from the ‘Related Documents’ section.

Proposed Changes

Note: Some headings are used only to keep the numbering consistent. (Proposed changes are in **blue**)

1 Scope

2 Normative references

2.1 Normative references

2.2 Approved references

2.3 References under development

2.4 IETF references

Copies of the following approved IETF standards may be obtained through the Internet Engineering Task Force (IETF) at the World Wide Web site <http://www.ietf.org>.

RFC 4303, *IP Encapsulating Security Payload (ESP)*
RFC 4306, *Internet Key Exchange (IKEv2) Protocol*

3 Definitions, acronyms, keywords, and conventions

3.1 Definitions

3.1.a **integrity check value (ICV):** a message authentication code used to validate the integrity of data passed across the service delivery subsystem. This is the term used by the Internet Engineering Task Force (IETF).

3.1.b **SA parameters**: The parameters stored by both an application client and a device server that are associated with one SA (see 3.1.m) and identified by a pair of SAIs. See SPC-4

3.1.c **Security association (SA):** A relationship and associated security processing between an application client and device server that is used to apply security functions (e.g., data integrity checking, data encryption) to data that is transferred in either direction. See SPC-4.

3.1.d **Security association index (SAI)**: A number representing the parameters for a security association as stored internally by the application client or device server. In other security models, this value is called the security parameters index (SPI). See SPC-4.

3.2 Acronyms

AES	Advanced Encryption Standard
FIPS	Federal Information Processing Standard
ICV	integrity check value
IETF	Internet Engineering Task Force
NIST	National Institute of Standards and Technology
RFC	(IETF) request for comment
SA	security association
SP	(NIST) special publication
SAI	security association identifier

4 General Concepts

4.1 Overview

4.2 Sequential-access device model

4.2.19 Archive tape and WORM mode

4.2.20 ~~Data encryption~~ Encryption of data on the volume

4.2.21 Encryption of information across the service delivery subsystem

4.2.21.1 Overview

The following subclauses describe cryptographic concepts needed for passing encrypted information packets over the service delivery subsystem. None of the concepts in the following subclauses refer to data stored on the media (see 4.2.20 for those concepts).

See SPC-4 for a generic description of a security association.

[Editor's Note: The reference to SPC-4 is only valid if 06-369 is approved]

NOTE - Many of these concepts are derived from the IETF's encapsulating security payload (ESP) protocol (see IETF RFC 4303 and RFC 4306).

4.2.21.2 ESP-SCSI

This subclause defines a method of encrypting parameter data called Encapsulating Security Payload for SCSI (ESP-SCSI). Before using ESP-SCSI, an application client and device server shall first establish a security association with a negotiated encryption algorithm and integrity algorithm (see SPC-4).

The negotiated encryption algorithm and integrity algorithm shall specify the following parameters:

a) Size of the initialization vector;
b) Size of the integrity check value;

The KEYMAT SA parameter (see SPC-4) shall consist of:

1) SK_ai – The shared key used by the integrity algorithm for traffic originating from the application client;
2) SK_ar – The shared key used by the integrity algorithm for traffic originating from the device server;
3) SK_ei – The shared key used by the encryption algorithm for traffic originating from the application client;
4) SK_er – The shared key used by the encryption algorithm for traffic originating from the device server;

Each shared key within KEYMAT shall be taken in order from the generated bits of the negotiated pseudo-random function (see SPC-4 and RFC 4306).

The size of each of the shared keys in KEYMAT is determined by the negotiated encryption algorithm and integrity algorithm.

Table E1 shows the format of ESP-SCSI applied to arbitrary parameter data.

Table E1

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)			SECURITY ASSOCIATION INDEX				
7								(LSB)
8	(MSB)			SEQUENCE NUMBER				
11								(LSB)
12	(MSB)			INITIALIZATION VECTOR (optional)				
12+k-1								(LSB)
12+k	(MSB)			ENCRYPTED DATA (variable length)				
m-s								(LSB)
m-s+1	(MSB)			INTEGRITY CHECK VALUE				
m								(LSB)

To process the information in Table E1, it is necessary to retrieve the following information from the security association (see SPC-4):

a) IV length, given as k in Table E1: The number of bytes within the IV is given by the negotiated security algorithm and referenced using the SAI; and

b) Integrity check value field length, given as s in Table A: The number of bytes within the integrity check value field is determined by the negotiated security algorithm

The SECURITY ASSOCIATION INDEX field contains the SAI value (see SPC-4) that identifies the security association that is used to encrypt the ENCRYPTED DATA field of this page.

The SEQUENCE NUMBER field contains the current SQN (see SPC-4) for the security association that is identified by the SAI. If the SQN is less than or equal to the SQN SA parameter in the SA identified by the SAI, then the device server shall terminate the command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID SA SEQUENCE NUMBER.

Editor's Note: INVALID SA SEQUENCE NUMBER is a new ASQ.

NOTE - The purpose of the DS_SQN is to prevent replay attacks (see SPC-4).

The INITIALIZATION VECTOR (optional) field contains an optional initialization vector used as an input into the encryption algorithm that encrypts the page. The security association specifies the size of the initialization vector, given as k in Table E1, based on the negotiated encryption algorithm (see SPC-4). The initialization vector is not encrypted.

The ENCRYPTED PAGE field contains the encrypted parameter data of a page referenced in Table 109 (e.g. Table 110 – Set Data Encryption page). The ENCRYPTED PAGE field shall not contain a Send Encapsulated Tape Data Encryption security protocol page (Table E1). The ENCRYPTED PAGE field shall be encrypted using the encryption algorithm referenced by the DS_SAI as negotiated during the creation of the security association (see SPC-4).

The INTEGRITY CHECK VALUE field contains an integrity check value that protects the following fields, in order:

1) SECURITY ASSOCIATION INDEX field;
2) SEQUENCE NUMBER field;
3) INITIALIZATION VECTOR field, if present; and
4) ENCRYPTED DATA field.

The device server shall verify the ICV field according to the algorithm negotiated during the creation of the security association. If the device server fails to validate the ICV field, then the device server shall terminate the command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID INTEGRITY CHECK VALUE. If the device server succeeds in validating the ICV, then the device server shall increment the DS_SQN SA parameter in the SA identified by the DS_SAI.

The size of the ICV field is given by the negotiated algorithm, and is shown as 's' in Table E1.

[Editor's note: INVALID INTEGRITY CHECK VALUE is a new sense code.]

5 Explicit address command descriptions for sequential-access devices
6 Implicit address command descriptions for sequential-access devices
7 Common command descriptions for sequential-access devices
8 Parameters for sequential-access devices
8.1 Diagnostic parameters
8.2 Log Parameters
8.3 Mode Parameters
8.4 Vital product data (VPD) parameters
8.5 Security protocol parameters

8.5.1 Security protocol overview
8.5.2 SECURITY PROTOCOL IN command specifying Tape Data Encryption security protocol
8.5.3 SECURITY PROTOCOL OUT command specifying Tape Data Encryption security protocol
8.5.3.1 SECURITY PROTOCOL OUT command specifying Tape Data Encryption security protocol overview

...

Table 109 – SECURITY PROTOCOL SPECIFIC field values

Code	Description	Reference
0000h – 000Fh	Reserved	
0010h	Set Data Encryption page	8.5.3.2
0011h	Send Encapsulated Tape Data Encryption security protocol page	8.5.3.3
0011h – FEFFh	Reserved	
FF00h – FFFFh	Vendor specific	

...

8.5.3.2 Set Data Encryption page

8.5.3.2.1 Set data encryption page format

Table Y1 shows the parameter list format of the Set Data Encryption page.

(Text skipped)

The KEY FORMAT field indicates the format of the value in the KEY field. Values for this field are described in Table Y5.

Table 114 – KEY FORMAT field values

Code	Description	Reference
00h	The KEY field contains the key to be used to encrypt or decrypt data.	8.5.3.2.2
01h	The KEY field contains a vendor specific key reference	8.5.3.2.3
02h	The KEY field contains the key wrapped by the device server's public key	8.5.3.2.4
03h	The KEY field contains a key that is encrypted using ESP	**Error! Reference source not found.**
04h – BFh	Reserved	N/A
C0h – FFh	Vendor specific	N/A

The KEY LENGTH field indicates the length of the key field in bytes.

(Move KEY-ASSOCIATED DATA DESCRIPTORS LIST text here)

8.5.3.2.2 Plaintext key format

If the KEY FORMAT field is 00h the KEY field contains the key in an algorithm specific format. Table Y8 defines the format of the key in the KEY field.

Table 115 – KEY field contents with KEY FORMAT field set to 00h

Bit / Byte	7	6	5	4	3	2	1	0
~~0~~20	(MSB)			KEY				
n								(LSB)

~~The KEY LENGTH field indicates the length of the key field in bytes~~.

8.5.3.2.3 Key reference format

~~If the KEY FORMAT field is 01h, the KEY field shall contain 8 bytes of T10 vendor identification (see SPC-4) followed immediately by a vendor specific key reference identifying the key to be used to encrypt or decrypt data. If the KEY field contains a vendor specific key reference that is unknown to the device server, the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to VENDOR SPECIFIC KEY REFERENCE NOT FOUND.~~

If the KEY FORMAT field is 01h, the KEY field shall contain the information as defined by Table Y8a.

Table Y8a – KEY field contents with KEY FORMAT field set to 01h

Bit / Byte	7	6	5	4	3	2	1	0
20	(MSB)			T10 VENDOR IDENTIFICATION				
27								(LSB)
28	(MSB)			VENDOR SPECIFIC KEY REFERENCE				
n								(LSB)

The T10 VENDOR IDENTIFICATION field shall contain 8 bytes of T10 vendor identification (see SPC-4).

The VENDOR SPECIFIC KEY REFERENCE field shall include vendor-specific information used for identifying the key that encrypts or decrypts data. If the VENDOR SPECIFIC KEY REFERENCE field contains a vendor specific key reference that is unknown to the device server, the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to VENDOR SPECIFIC KEY REFERENCE NOT FOUND.

8.5.3.2.4 Key Wrapped by Device Server's Public Key

8.5.3.2.5 Key encrypted using ESP-SCSI

If the KEY FORMAT field of Table Y1 is 03h, the KEY field shall contain a key that is encrypted using ESP-SCSI (see 4.2.21.2).

8.5.3.3 Send Encapsulated Tape Data Encryption security protocol page

Table A specifies the format of the Send Encapsulated Tape Data Encryption security protocol page

Table A – Send Encapsulated Tape Data Encryption security protocol page

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)			PAGE CODE (0011h)				
1								(LSB)
2	(MSB)			PAGE LENGTH (m-3)				
3								(LSB)
4	(MSB)			DEVICE SERVER SECURITY ASSOCIATION INDEX				
7								(LSB)
8	(MSB)			DEVICE SERVER SEQUENCE NUMBER				
15								(LSB)
16	(MSB)			INITIALIZATION VECTOR (optional)				
16+k-1								(LSB)
16+k	(MSB)			ENCRYPTED PAGE (variable length)				
m-s								(LSB)
m-s+1	(MSB)			INTEGRITY CHECK VALUE				
m								(LSB)

To process the information in Table A, it is necessary to retrieve the following information from the security association (see SPC-4):

c) IV length, given as k in Table A: The number of bytes within the IV is given by the negotiated security algorithm and referenced using the SAI.

d) Integrity check value field length, given as s in Table A: The number of bytes within the integrity check value field is determined by the negotiated security algorithm

The PAGE CODE field shall contain the value 0011h.

The PAGE LENGTH field indicates the number of bytes of parameter data to follow. If the page length value results in the truncation of any field, the device server shall terminate the command with CHECK CONDITION status,

with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN PARAMETER LIST.

The DEVICE SERVER SECURITY ASSOCIATION INDEX field contains the DS_SAI value (see SPC-4) that identifies the security association for the device server that is used to encrypt the ENCRYPTED PAGE field of this page.

The DEVICE SERVER SEQUENCE NUMBER field contains the current DS_SQN (see SPC-4) for the security association that is identified by the DS_SAI. If the DS_SQN is less than or equal to the DS_SQN SA parameter in the SA identified by the DS_SAI, then the device server shall terminate the command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID SA SEQUENCE NUMBER.

Editor's Note: INVALID SA SEQUENCE NUMBER is a new ASQ.

NOTE - The purpose of the DS_SQN is to prevent replay attacks (see SPC-4).

The INITIALIZATION VECTOR (optional) field contains an optional initialization vector used as an input into the encryption algorithm that encrypts the page. The security association specifies the size of the initialization vector, given as k in Table A, based on the negotiated encryption algorithm (see SPC-4). The initialization vector is not encrypted.

The ENCRYPTED PAGE field contains the encrypted parameter data of a page referenced in Table 109 (e.g. Table 110 – Set Data Encryption page). The ENCRYPTED PAGE field shall not contain a Send Encapsulated Tape Data Encryption security protocol page (Table A). The ENCRYPTED PAGE field shall be encrypted using the encryption algorithm referenced by the DS_SAI as negotiated during the creation of the security association (see SPC-4).

The INTEGRITY CHECK VALUE field contains an integrity check value that protects the following fields, in order:

5) DEVICE SERVER SECURITY ASSOCIATION INDEX field;
6) DEVICE SERVER SEQUENCE NUMBER field;
7) INITIALIZATION VECTOR field, if present; and
8) ENCRYPTED PAGE field.

The device server shall verify the ICV field according to the algorithm negotiated during the creation of the security association. If the device server fails to validate the ICV field, then the device server shall terminate the command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID INTEGRITY CHECK VALUE. If the device server succeeds in validating the ICV, then the device server shall increment the DS_SQN SA parameter in the SA identified by the DS_SAI.

The size of the ICV field is given by the negotiated algorithm, and is shown as ‘s’ in Table A.

Editor’s note: INVALID INTEGRITY CHECK VALUE is a new sense code.