To: INCITS T10 Committee

From: Paul Entzel, Quantum

Date: 16 March 2006

Document: T10/05-446r8

Subject: SSC-3: Add commands to control data encryption

1 Revision History

Revision 0:
Posted to the T10 web site on 16 November 2005.

Revision 1:
Posted to the T10 web site on 12 December 2005. Includes:

1. In 3.1.X, change “ciphering process was performed by the device server” to “ciphering process was performed by a device server”.
2. In 7.X.4, change AK_C to MAC_C and the paragraph that describes it.
3. Add CKOD bit to Set Data Encryption page.
4. Add microcode update and other vendor specific events to the list of events that clear a key.
5. Add a paragraph to 4.2.19.5 describing the loss of a key due to a vendor specific event.
6. Change GLOBAL scope to ALL I_T NEXUS.
7. Added key generation concept (4.2.18.6 and 4.2.19.9) and removed the ENCRYPTE bit from the WRITE(6) and WRITE(16) commands.
8. Added descriptors for key-associated data (subclause 8.5). Add these descriptors to the Data Encryption Status page and the Set Data Encryption page.
9. Added Next Block Encryption Status page to report encryption status and KAD data for the next block to read (still looking for a better name for this page).
10. Added ALGORITHM INDEX field to the Data Encryption Status page.
11. Expand the ENCRYPT and DECRYPT fields (renamed ENCRYPTION MODE and DECRYPTION MODE) in the Set Data Encryption page and the Data Encryption Status page to 4 bits and add multiple modes from HP’s strawman proposal (with some name changes).
12. Add LOCK bit to Set Data Encryption and Data Encryption Status pages and a model clause to describe it.
13. Add maximum key-associated data lengths to the algorithm descriptors.
14. Added acronyms for A-KAD and U-KAD.
15. Added a key descriptor for the application client to set the nonce and for the device server to report this capability/requirement.

Revision 2:
Updated with comments from the SSC-3 WG meeting on 10 January 2006. Posted to the T10 web site on 17 January 2006. Includes these major changes:

1. Corrected various spelling and grammatical errors.
2. Added a paragraph in 4.2.19.3 describing behavior when an encrypted block fails to authenticate.
3. Added test to 4.2.19.5 detailing the data to be saved from Set Encryption Mode page processing for each scope.
4. In 4.2.19.9 corrected several instances of DATA SECURITY IN to DATA SECURITY OUT.
5. Swapped several bytes in the Data Encryption Status page to better align with the Set Encryption Mode page.
6. Fixed the alignment in the Next Block Encryption Status page.
7. Add code value of 5h to Table N3.
8. Added Vendor Specific code ranges to the KEY FORMAT and KEY DESCRIPTOR TYPE tables.
9. Removed the AUTH and VALID bits in the DATA SECURITY descriptor format table with a multiple bit field called AUTHENTICATED. Added table D3 to describe this field’s values.

Revision 3:
Posted to the T10 web site on 19 January 2006. Converted from DATA SECURITY IN and DATA SECURITY OUT commands (new commands defined by previous revisions of this proposal) to SECURITY PROTOCOL IN and SECURITY PROTOCOL OUT commands.

Revision 4:
Posted to the T10 web site on 31 January 2006. Contain numerous editorial changes in addition to the following technical changes:

1. Added a definition for the term “data encryption parameters” and switched to using this term in place of “encryption key and mode” and other similar terms.
2. Replaced “hard reset event” with “power-on” as a reason to clear a key.
3. Moved the order of precedence for selecting data encryption parameters from subclause 8.5.3.2 to the Managing Keys model clause (4.2.19.5).
4. Fixed the description of Key Generation to include size, initialization, and rollover.
5. Changed the name of the Next Block Status page to be Next Block Encryption Status page.
6. Added several editor’s notes in purple indicating comments received that should be discussed more.
7. Modified the description of the PUBLIC scope to ignore all but SCOPE and LOCK.
8. Add READ REVERSE(6) and READ REVERSE(16) to the list of commands that will decrypt.
9. Added a sentence to 8.5.3.2 to reference 4.2.19.9 for a description of the LOCK bit.
10. Added Supported Key Format page and removed KBR_C bit from Data Encryption Algorithms descriptor.
11. Incorporate 06-050r2 with appropriate changes.

Revision 5 (we’re up to 30 pages now):
Posted to the T10 web site on 13 February 2006: Contains changes recommended by the SSC-3 working group during the 8 February conference call:

1. Moved the list of things that can cause encryption to be disabled from 4.2.19.2 to 4.2.19.5 and reworded it to things that can cause a data encryption parameter resource to be released.
2. Reworded 4.2.19.2 to indicate encryption is managed on a per I_T Nexus basis.
3. Added a reference for DED_C in 4.2.19.3 and a note that “it is bad to not be able to distinguish encrypted data from non-encrypted data”. Also added a paragraph describing the order of precedence in reporting bad key over authentication failure.
4. Changed 4.2.19.4 to no longer describe how to prevent exhaustive search attacks.
5. Moved the stuff about what is saved on a per I_T Nexus basis into its own sub-clause (4.2.19.6). Added some test to describe the scope for the I_T Nexus.
6. Combined all of the data encryption parameter structures into one and added a paragraph to describe resource limitations.
7. Changed “key generation” to “key instance counter” and made it 32 bits.
8. Added some reserved space to the Data Encryption Capabilities page.
9. Changed the SCOPE and SOURCE fields in the Data Encryption Status page to I_T NEXUS SCOPE and KEY SCOPE respectively.
10. Added RECOVER BUFFERED DATA to the list of commands that decrypt.
11. Added Data Encryption Management Capabilities page.
12. Added multiple purple editors’ notes to cover the areas we still need to address.

Revision 6:
Posted to the T10 web site on 20 February 2006: Contains changes recommended by the SSC-3 working group during the 17 February conference call:

1. Use title capitalization for the new protocol name Tape Data Encryption.
2. In subclause 4.2.19.5, remove the list of key formats and changed the text to require one of the standardized formats be supported.
3. Made support for ALL I_T NEXUS and PUBLIC scoped mandatory.
4. Corrected several instances of Set Encryption Parameters being used instead of Set Data Encryption for the control page name.

5. Added a sentence that default parameters shall be used for an I_T Nexus that has a scope of LOCAL and loses its parameters.
6. Added subclause 4.2.19.10 defining KAD (from an email by Chris Williams).
7. Added some reserved space in table S1 to accommodate 06-051 and any future needs.
8. Added a paragraph to 8.5.3.2 describing the KEY field for a key format of 00h (plain text).
9. Removed the requirement to check the T10 vendor ID when the key format is 01h (key by reference).
10. Removed the blue editor's notes. These can now be found in T10/06-108 as a proposal against SPC-4.

Revision 7:
Updated with comments from the SSC-3 WG meeting on 7 March 2006. Posted to the T10 web site on 8 March 2006. Includes these major changes:

1. Added definitions for Nonce and Message Authentication Code.
2. Renamed "Tape Data Encryption protocol" to "Tape Data Encryption security protocol".
3. Replaced many instances of "able to" with "capable of".
4. Replaced "authenticate" with "validate the integrity of" in many places.
5. Removed the second paragraph in subclause 4.2.19.4.
6. Split the CKROL case for clearing the keys into two list items to better describe the difference between local and shared keys.
7. Changed "clear all memory" to "change the contents of all memory: in several places.
8. Modified the exception handling paragraphs in 4.2.19.5 to be unordered lists for clarification.
9. Added a paragraph to 4.2.19.6 specifying the default data encryption scope and lock value in the saved I_T Nexus parameters.
10. Remove the introductory paragraph in 4.2.19.10.
11. Renumbered 4.2.19.10 (second instance) to 4.2.19.11. Removed most of the subclause.
12. In the Data Encryption Algorithm identifiers, removed the ALGORITHM NAME LENGTH and ALGORITHM NAME fields with an ENCRYPTION ALGORITHM IDENTIFIER field and added a paragraph to see SPC-4. Now we just need to get a table added to SPC-4 and we are home free.
13. Replaced "dismount" with "de-mount" in several places.

Revision 8:
Posted to the T10 web site on 16 March 2006: Contains changes recommended by the SSC-3 working group during the 16 March 2006 conference call:

1. Added Reservation Loss definition and references to it.
2. Added CKORSC bit to Set Data Encryption page. Added CKORSC_C bit to Data Encryption Management Capabilities page.
3. Removed table P1 and the sentence that referenced it.
4. Reconstructed tables C1 and D1 to show format of the page codes.
5. Changed the name of the IV_EOU bit to IV_EBU.
6. Shuffled the bits around in byte 4 and 5 of the Data Encryption Management Capabilities page.
7. Expanded the ENCRYPTION MODE and DECRYPTION MODE fields from 4 bits each to 8 bits each in the Data Encryption Status page and the Set Data Encryption page.
8. Modified many of the field descriptions in the Data Encryption Status page to reference the Data Encryption Parameters structure and referenced the model clause.
9. Modified tables N2 and N3 to reference the LOGCIAL OBJECT NUMBER field instead of the next block. Added a code to each table for objects that are not logical blocks. Added "the device server has determined…" to many of the descriptions.
10. In the Set Data Encryption page, increased the ENCRYPTION MODE and DECRYPTION MODE fields to 8 bits and moved the CK bits into their own byte. Shuffled some fields around to make this fit.
11. Removed the reference to metadata from table Y4.
12. Added table Y8 to describe the key format.
13. Removes the remaining purple editor's notes.

14. Added a KEY SIZE field to the algorithm descriptors since the algorithm identifiers used by IETF do not contain that information.

2 General

A great deal of interest has been expressed to add access security to data recorded on removable medium in the wake of several high profile cases where medium containing sensitive data has been lost or stolen. Many back-up applications support adding passwords to data sets and/or encrypting the data. The password scheme does not protect from the data being recovered using a different application that understands the format but does not honor the password protection. The encryption approach is safer, but introduces significant performance degradation when used.

The major disadvantage to encrypting the data concerns the performance impact. Software algorithms to encrypt the data are available, but they take significant time to perform and this reduces the available performance. Also, since the encryption process removes redundancy in the data, attempts to compress the data after encryption are usually unsuccessful. In order to regain the performance boost from compression, the compression must be done before encryption, so it must be done in software. This adds even more time overhead, reducing the performance even more.

This proposal provides a method where encryption could be moved into the device, solving the problems described above. This proposal provides a method to control these features by introducing a new protocol for use with the SECURITY PROTOCOL IN and SECURITY PROTOCOL OUT commands. The new protocol is named “Tape Data Encryption”.

The SECURITY PROTOCOL IN command using Tape Data Encryption security protocol can request several reports containing supported features, enabled modes, and status.

The SECURITY PROTOCOL OUT command using Tape Data Encryption security protocol is used to set encryption keys and modes of protection.

Perhaps the most difficult problem to solve with the addition of this interface is how to deal with multiple initiators. One would think that the parameters controlling encryption and decryption would be I_T nexus specific, and that they would not be used without reservations in place. However, that won’t work in many environments:

1. If the target port is behind a protocol bridge, the device server can not tell one initiator from another. In this environment, parameters that are normally per I_T nexus are shared at the target device level and the protocol bridge is required to sort out the individual initiator ports.
2. To support copy managers, we either need to add the ability to pass encryption modes and keys via the EXTENDED COPY command, have the ability to pass the set-up from one I_T Nexus to another, or have the ability to share set-up.

Future enhancements to this feature may include adding a method to encrypt the data encryption key before passing it to the device server. The addition of this single feature was proving to be significantly more complex than what was already contained in the proposal. We decided to postpone discussion of this feature until a later proposal. This proposal establishes plenty of reserved fields and page codes so that it can be added without breaking everything else.

3 Changes to SSC-3

3.1 Additions to the definition clause (3)

3.1.W Data encryption parameters – A set of parameters accessible through the Set Data Encryption page (see 8.5.3.2) that controls the data encryption and decryption process in the device server. See 4.2.19.7.

3.1.X Encrypted Block – A Logical Block in which the data has been subjected to a ciphering processes by the device server. Within this standard, a logical block is only considered encrypted if the ciphering process was performed by a device server.

3.1.A Message Authentication Code – A short piece of information used to validate the integrity of data.

3.1.Z Nonce – A value that is used one and only one time and thus uniquely identifies a single instance of something (e.g., an initialization vector for an encryption algorithm).

3.1.B Reservation Loss – An event caused by the release of a reserve/release method reservation (see SPC-2) or by the transition within the device server from the state where a persistent reservation holder exists to the state where a persistent reservation holder does not exist (see SPC-4).

3.1.Y Unencrypted Block – A Logical Block in which the data has not been subjected to a ciphering process by the device server.

In subclause 3.2 add the following acronyms:

A-KAD	Authenticated key-associated data
MAC	Message authentication code
U-KAD	Unauthenticated key-associated data

3.2 Addition to the model clause (4)

Add the following subclause to clause 4:

4.2.19 Data Encryption

4.2.19.1 Data Encryption overview

A device compliant with this standard may contain hardware or software that is capable of encrypting and decrypting the data within logical blocks to provide security against unauthorized access to that data. The SECURITY PROTOCOL IN and SECURITY PROTOCOL OUT commands specifying the Tape Data Encryption security protocol (see Table P1) provide a means for the application client to monitor and control the encryption and decryption processes within the device server. A device server that supports the SECURITY PROTOCOL OUT command shall also support the SECURITY PROTOCOL IN command.

The SECURITY PROTOCOL OUT command specifying the Tape Data Encryption security protocol is used to set data encryption parameters. The SECURITY PROTOCOL IN command specifying the Tape Data Encryption security protocol is used to discover the type of data security features supported by the device server, the current configuration of data security features, and status of the encryption and decryption processes.

4.2.19.2 Encrypting data on the medium

The application client controls the data encryption process by use of the SECURITY PROTOCOL OUT command specifying the Tape Data Encryption security protocol. Data encryption shall be managed

within the device server on a per I_T Nexus basis. The data encryption process is enabled for an I_T Nexus upon successful completion of a SECURITY PROTOCOL OUT command that sends a Set Data Encryption page (see 8.5.3.2) with the ENCRYPTION MODE field set to ENCRYPT and with a valid key. If the data encryption scope parameter for an I_T Nexus is set to PUBLIC (See 4.2.19.6), the data encryption process may be enabled by another I_T Nexus that establishes a set of data encryption parameters with a key scope of ALL I_T NEXUS or RESERVATION GROUP (see 4.2.19.7).

If data encryption is enabled for an I_T Nexus, all data received by the device server from that I_T Nexus as part of a WRITE(6) or WRITE(16) command shall be encrypted before being recorded on the medium. Filemarks shall not be encrypted.

4.2.19.3 Reading encrypted data on the medium

A volume may contain no encrypted blocks, all encrypted blocks, or a mixture of encrypted blocks and unencrypted blocks. The fact that blocks are encrypted shall not alter space or locate operations.

A device server that supports encryption should be capable of distinguishing encrypted blocks from unencrypted blocks. The device server reports its capability of distinguishing encrypted blocks from unencrypted blocks through the DED_C bit in the Data Encryption Algorithm descriptor (see 8.5.2.4). If the device server is capable of distinguishing encrypted blocks from unencrypted blocks, an attempt to read or verify an encrypted block when the decryption mode is set to DISABLED shall cause the device server to terminate the command with CHECK CONDITION status, with the sense key set to DATA PROTECT, and the additional sense code set to UNABLE TO DECRYPT DATA. The device server shall establish the logical position at the BOP side of the encrypted block.

If the device server is capable of distinguishing encrypted blocks from unencrypted blocks and the decryption mode is set to DECRYPT or RAW, an attempt to read or verify an unencrypted block shall cause the device server to terminate the command with CHECK CONDITION status, with the sense key set to DATA PROTECT, and the additional sense code set to UNENCRYPTED DATA ENCOUNTERED WHILE DECRYPTING. The device server shall establish the logical position at the BOP side of the unencrypted block.

NOTE E1: It is possible for a device server that is not capable of distinguishing encrypted blocks from unencrypted blocks to decrypt data that was not encrypted. Application clients are responsible for checking the integrity of the data in this environment.

A device server that supports encryption and has been configured to decrypt the data may be capable of determining that the encryption key is correct for an encrypted block. If the device server is capable of determining that the encryption key is correct, an attempt to read or verify an encrypted block when the decryption mode is set to either DECRYPT or MIXED but the encryption key is incorrect for the encrypted block shall cause the device server to terminate the command with CHECK CONDITION status, with the sense key set to DATA PROTECT, and the additional sense code set to INCORRECT DATA ENCRYPTION KEY. The device server shall establish the logical position at the BOP side of the encrypted block.

A device server that supports encryption and has been configured to decrypt the data may be capable of validating the integrity of the data after decrypting it (i.e., that the decrypted data matches the data that was encrypted). If the device server is capable of validating the integrity of the data after decrypting it, an attempt to read or verify an encrypted block when the decryption mode is set to either DECRYPT or MIXED but the data fails the integrity validation process shall cause the device server to terminate the command with CHECK CONDITION status, with the sense key set to DATA PROTECT, and the additional sense code set to CRYPTOGRAPHIC INTEGRITY VALIDATION FAILED. The device server shall establish the logical position at the BOP side the encrypted block.

A device server that is capable of distinguishing encrypted blocks from unencrypted blocks and has been configured to decrypt the data should perform at least one of the following for each encrypted block that is decrypted:

a) determine if the encryption key is correct for the encrypted block; or
b) validate the integrity of the data after decrypting it.

A device server that is capable of both determining if the encryption key is correct for the encrypted block and validating the integrity of the data after decrypting it shall:

1. determine if the encryption key is correct for the encrypted block; and
2. validate the integrity of the data.

4.2.19.4 Exhaustive-search attack prevention

To prevent an exhaustive-search attack from discovering the encryption key, the device server should provide a mechanism to prevent unlimited attempts at setting a data encryption key and then attempting to read the data. The use of such a mechanism may protect against an encryption algorithm being broken due to undiscovered mathematical weaknesses in the encryption algorithm.

If the device server has reached its limit on failed attempts to set the data encryption key and decrypt data, it shall disable decryption for all I_T nexuses. All subsequent SECURITY PROTOCOL OUT commands specifying the Tape Data Encryption security protocol and with the SP_SPECIFIC field set to Set Data Encryption page with the DECRYPT field or ENCRYPT field set to any value other than DISABLE shall be terminated with CHECK CONDITION status, with the sense key set to DATA PROTECT, and the additional sense code set to DATA DECRYPTION KEY FAIL LIMIT REACHED. This condition shall persist until the volume is de-mounted from the device or a hard reset condition occurs.

4.2.19.5 Managing keys within the device server

To increase the security of keys, the data encryption parameters are volatile in the device server and the data encryption keys are never reported to an application client. The device server also may have limited resources for storage of keys.

A device server that supports encryption shall support at least one of the key formats that are defined in this standard (see table Y5).

A vendor specific key reference is an identifier that is associated with a specific key. The method by which keys and their associated vendor specific key references are made available to the device server is outside the scope of this standard. A device server that supports passing keys by vendor specific key reference shall include the code for vendor specific key reference format (see table Y5) in the SUPPORTED KEY FORMATS LIST field in the Supported Key Formats Page (see 8.5.2.5).

The device server shall release the resources used to save a set of data encryption parameters under the following conditions:

a) the CKOD bit is set to one in the saved data encryption parameters and the volume is de-mounted;

b) the CKORL bit is set to one and the key scope is set to LOCAL in the saved data encryption parameters and the I_T nexus that established the set of data encryption parameters loses its reservation;

c) the CKORL bit is set to one and the key scope is set to RESERVATION GROUP or ALL I_T NEXUS in the saved data encryption parameters and the device server experiences a reservation loss (see 3.1.B);

d) the CKORSC bit is set to one in the saved data encryption parameters and the device server processes a PERSISTENT RESERVATION OUT command that changes the scope of the persistent reservation;

e) a microcode update is performed on the device;

f) a power on condition occurs; or

g) other vendor specific events.

If a device server processes a Set Data Encryption page with the ENCRYPTION MODE field set to DISABLE and DECRYPTION MODE field set to DISABLE or RAW, the device server shall:

a) release any resources that it had allocated to store data encryption parameters for the I_T nexus associated with the SECURITY PROTOCOL OUT command and shall change the contents of all memory containing a key value associated with the data encryption parameters that are released; and

b) establish a unit attention condition with the additional sense of DATA ENCRYPTION PARAMETERS CHANGED BY ANOTHER INITIATOR for all other I_T nexus that are affected by the loss of the key, (i.e., any I_T nexus that is using a data encryption scope of PUBLIC and sharing the key.)

If a device server processes a Set Data Encryption page that includes a key, the device server shall:

a) release all resources that it had allocated to store a key value set by a previous SECURITY PROTOCOL OUT command from that I_T nexus and shall change the contents of all memory containing a key value associated with the data encryption parameters that are released; and

b) establish a unit attention condition with the additional sense of DATA ENCRYPTION PARAMETERS CHANGED BY ANOTHER INITIATOR for all other I_T nexus that is affected by the change of the key (i.e., any I_T nexus that is using a data encryption scope of PUBLIC and sharing the key).

A device server shall save at most one set of data encryption parameters with a key scope of ALL I_T NEXUS. If a device server processes a Set Data Encryption page with the SCOPE field set to ALL I_T NEXUS, the device server shall:

a) release any resources that it had allocated to store data encryption parameters set by a previous Set Data Encryption page with a scope value of ALL I_T NEXUS and shall change the contents of all memory containing a key value associated with the data encryption parameters that are released; and

b) establish a unit attention condition with the additional sense of DATA ENCRYPTION PARAMETERS CHANGED BY ANOTHER INITIATOR for all other I_T nexus that are affected by the change of the key (i.e. any I_T nexus that is using a data encryption scope of PUBLIC and sharing the key.)

A device server shall save at most one set of data encryption parameters with a key scope of RESERVATION GROUP. If a device server processes a Set Data Encryption page with the SCOPE field set to RESERVATION GROUP, the device server shall:

a) release any resources that it had allocated to store data encryption parameters set by a previous Set Data Encryption page with a scope value of RESERVATION GROUP and shall change the contents of all memory containing a key value; and

b) establish a unit attention condition with the additional sense of DATA ENCRYPTION PARAMETERS CHANGED BY ANOTHER INITIATOR for any other I_T nexus that is affected by the change of the key, (i.e., any I_T nexus that is using a data encryption scope of PUBLIC and sharing the key.)

If a vendor specific event occurs that changes or clears a set of data encryption parameters, the device server shall establish a unit attention condition with the additional sense of DATA ENCRYPTION PARAMETERS CHANGED BY VENDOR SPECIFIC EVENT for any I_T nexus that is affected by the change of the key.

4.2.19.6 Saved information per I_T Nexus

If the device server supports data encryption it shall save the following information on a per I_T nexus basis:

a) data encryption scope;
b) lock;
c) key instance counter value at lock; and
d) key instance counter value assigned to the last key established by a Set Data Encryption page for this I_T Nexus with a scope value of LOCAL;

The set of possible data encryption scope values for an I_T nexus is limited to:

a) PUBLIC;
b) LOCAL;
c) RESERVATION GROUP; or
d) ALL I_T NEXUS

If an I_T Nexus data encryption scope is set to PUBLIC it indicates the device server does not have a saved set of data encryption parameters that were established by that I_T Nexus. Device servers that support encryption shall support an I_T Nexus data encryption scope of PUBLIC.

A device server shall set the data encryption scope for an I_T Nexus to LOCAL when it successfully completes the processing of a Set Data Encryption page with a scope of LOCAL from that I_T Nexus. The device server shall only use the data encryption parameters established by the Set Data Encryption page with a scope of LOCAL for processing commands from the I_T Nexus that established the parameters. A device server shall revert to using default data encryption parameters for an I_T Nexus that is configured with a data encryption scope of LOCAL if the resources used to save the data encryption parameters for the I_T Nexus are released.

A device server shall set the data encryption scope for an I_T Nexus to RESERVATION GROUP when it successfully completes the processing of a Set Data Encryption page with a scope value of RESERVATION GROUP from that I_T Nexus. At most, one I_T Nexus shall be assigned the data encryption scope of RESERVATION GROUP. If the device server releases resources used to store a set of data encryption parameters with a key scope of RESERVATION GROUP, it shall change the data encryption scope for the I_T Nexus that established that set of data encryption parameters to PUBLIC.

A device server shall set the data encryption scope for an I_T Nexus to ALL I_T NEXUS when it successfully completes the processing of Set Data Encryption page with a scope value of ALL I_T NEXUS from that I_T Nexus. At most, one I_T Nexus shall be assigned the data encryption scope of ALL I_T NEXUS. If the device server releases resources used to store a set of data encryption parameters with a key scope of ALL I_T NEXUS, it shall change the data encryption scope for the I_T Nexus that established that set of data encryption parameters to PUBLIC. Device servers that support encryption shall support an I_T Nexus data encryption scope of ALL I_T NEXUS.

By default, the device server shall set the saved I_T Nexus parameters data encryption scope value to PUBLIC and lock value to zero.

4.2.19.7 Data encryption parameters

A device server that supports data encryption shall have the ability to save the following information as a set of data encryption parameters when a Set Data Encryption page is processed:

a) For SCSI transport protocols where initiator port names are required, the initiator port name; otherwise, the initiator port identifier;
b) indication of the target port through which the data encryption parameters were established;
c) key scope;
d) encryption mode;
e) decryption mode;
f) key;
g) algorithm index;
h) key instance counter;
i) CKOD;
j) CKORL;
k) U-KAD;
l) A-KAD; and
m) nonce.

A device server may have limited resources for storage of sets of data encryption parameters (i.e., it may not have enough resources to store a unique set of data encryption parameters for every I_T Nexus that it is capable of managing). A device server may release a previously established set of data encryption parameters when a Set Data Encryption page is processed and there are no unused resources available. The method of choosing which set of data encryption parameters to release is vendor specific. If the device server does release a previously established set of data encryption parameters to free the resource, it shall establish a unit attention condition for every affected I_T Nexus (see 4.2.19.5).

If the device server supports an encryption key scope of ALL I_T NEXUS, it shall have resources to save one set of data encryption parameters with this scope.

If the device server supports an encryption key scope of RESERVATION GROUP, it shall have resources to save one set of data encryption parameters with this scope.

If the device server supports an encryption key scope value of LOCAL, it shall have resources to save one or more sets of data encryption parameters with this scope.

The data encryption parameters that shall be used for an I_T nexus shall be established by the following order of precedence:

a) If the data encryption scope for the I_T nexus is set to LOCAL, RESERVATION GROUP, or ALL I_T NEXUS (see 4.2.19.6), the data encryption parameters set by the last Set Data Encryption page from that I_T Nexus; or

b) If the data encryption scope for the I_T nexus is set to PUBLIC:

    1) If there is a reservation in effect for the logical unit, the data encryption parameters that have been saved by the device server with a key scope of RESERVATION GROUP if any data encryption parameters have been saved with this key scope;

    2) the data encryption parameters that have been saved by the device server with a key scope of ALL I_T NEXUS if any data encryption parameters have been saved with this key scope; or

    3) the default data encryption parameters.

4.2.19.8 Key instance counter

The device server shall keep a counter for each key that it is managing called the key instance counter. All key instance counters shall be set to zero when a power on condition occurs. Any other event that sets, clears, or changes a parameter in a set of data encryption parameters shall cause the key instance counter for that set of data encryption parameters to be incremented. The value of the key instance counter associated with the currently selected key for an I_T Nexus is reported in the Data Encryption Status page of SECURITY PROTOCOL IN command. The key instance counters are 32 bits and shall roll over to zero when incremented past their maximum value.

4.2.19.9 Encryption mode locking

There are conditions outside of the control of an application client which cause the device server to release the resources used to save the data encryption parameters (see 4.2.19.5) or change the data encryption parameters used to control the encryption of data. Each of these conditions cause the device server to establish a unit attention condition to report the change of operating mode, but the unit attention condition may not always be reported to the application client through protocol bridges and driver stacks.

The LOCK bit in the Set Data Encryption page is set to one to lock the I_T Nexus that issued the SECURITY PROTOCOL OUT command to the set of data encryption parameters established at the completion of the processing of the command. The I_T nexus remains locked to that set of data encryption parameters and key instance counter value until a hard reset condition occurs or another SECURITY PROTOCOL OUT command including a Set Data Encryption page from the same I_T Nexus is processed.

If the device server processes a WRITE(6) or WRITE(16) command for an I_T Nexus that is locked to a set of data encryption parameters and key instance counter, and the key instance counter value has changed since the time it was locked, the device server shall terminate the command with CHECK CONDITION status, with the sense key set to DATA PROTECT, and the additional sense code set to DATA ENCRYPTION KEY INSTANCE COUNTER HAS CHANGED. All subsequent WRITE(6) and WRITE(16) commands shall also be terminated this way until a hard reset condition occurs or a SECURITY PROTOCOL OUT command including a Set Data Encryption page from the same I_T Nexus is processed.

4.2.19.10 Nonce generation

For a given encryption algorithm, the device server may:

a) not require a nonce value;
b) generate its own nonce value;
c) require a nonce value or part of the nonce value be provided by the application client; or
d) be configurable with respect to the source of the nonce value.

The device server reports its capability with respect to nonce values in the Encryption Algorithm Descriptions (see 8.5.3.2). If the device server reports that it requires a nonce value from the application client and a Set Data Encryption page is processed that does not include a nonce value descriptor, the device server shall terminate the command with CHECK CONDITION, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INCOMPLETE KEY-ASSOCIATED DATA SET.

4.2.19.11 Unauthenticated Key Associated Data (U-KAD) and Authenticated Key Associated Data (A-KAD)

Some encryption algorithms allow or require the use of additional data which is associated with the key and the plaintext, but which is not encrypted. It may be authenticated by being included in the message authentication code (MAC) calculations for the encrypted plaintext if such an MAC exists, or non-authenticated by not being included in these calculations.

Note ZZ: a key-identifier or key reference may be stored in the U-KAD or A-KAD

The U-KAD field is provided for applications that do not require the KAD to be protected by an MAC.

3.3 Changes to the explicit command set clause (5)

3.3.1 Addition to subclause 5.1 Summary of commands for explicit address mode

In table 13, add the following commands:

Command Name	Type	Opcode	Synchronization Operation Required	Reference
SECURITY PROTOCOL IN	O	A2h	No	SPC-4
SECURITY PROTOCOL OUT	O	B5h	No	SPC-4

3.4 Changes to the implicit command set clause (6)

3.4.1 Addition to subclause 6.1 Summary of commands for implicit address mode

In table 20, add the following commands:

Command Name	Type	Opcode	Synchronization Operation Required	Reference
SECURITY PROTOCOL IN	O	A2h	No	SPC-4
SECURITY PROTOCOL OUT	O	B5h	No	SPC-4

3.5 Additions to the Parameters for sequential-access devices clause (8)

8.5. Security protocol parameters

8.5.1 Security protocol overview

This subclause describes the protocols, pages, and descriptors used by sequential-access devices with the SECURITY PROTOCOL IN and SECURITY PROTOCOL OUT commands (see SPC-4).

8.5.2 SECURITY PROTOCOL IN command specifying Tape Data Encryption security protocol

8.5.2.1 SECURITY PROTOCOL IN command specifying Tape Data Encryption security protocol overview

The SECURITY PROTOCOL IN command (see SPC-4) specifying Tape Data Encryption security protocol (i.e., 20h) requests the device server to return information about the data security methods in the device server and on the medium. The command supports a series of pages that are requested individually. An application client requests a page by using a SECURITY PROTOCOL IN command with the SECURITY PROTOCOL field set to Tape Data Encryption security protocol and the SP_SPECIFIC field set to the page code requested.

The SP_SPECIFIC field (see Table B1) indicates the type of report that the application client is requesting.

Table B1 – SP_SPECIFIC **field values**

Code	Description	Reference
0000h	Supported Tape Data Encryption In pages page	8.5.2.2
0001h	Supported Tape Data Encryption Out pages page	8.5.2.3
0002h – 000Fh	Reserved	
0010h	Data Encryption Capabilities page	8.5.2.4
0011h	Supported Key Formats page	8.5.2.5
0012h	Data Encryption Management Capabilities page	8.5.2.6
0013h – 001Fh	Reserved	
0020h	Data Encryption Status page	8.5.2.7
0021h	Next Block Encryption Status page	8.5.2.8
0013h - FFFFh	Reserved	

The ALLOCATION LENGTH field specifies the maximum number of bytes that the device server may return (see SPC-4).

8.5.2.2 Supported Tape Data Encryption In pages page

Table C1 shows the format of the Supported Tape Data Encryption In pages page.

Table C1 – Supported Tape Data Encryption In pages page

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)	PAGE CODE (0000h)						
1								(LSB)
2	(MSB)	PAGE LENGTH (n-3)						
3								(LSB)
	Supported tape data encryption in page code list							
4	(MSB)	Supported tape data encryption in page code (first)						
5								(LSB)
	:							
n-1	(MSB)	Supported tape data encryption in page code (last)						
n								(LSB)

The SUPPORTED PAGE LIST field shall contain a list of all of the pages that the device server supports for the SECURITY PROTOCOL IN command specifying the Tape Data Encryption security protocol in ascending order beginning with page code 0000h.

8.5.2.3 Supported Tape Data Encryption Out pages page

Table D1 shows the format of the Supported Tape Data Encryption Out pages page.

Table D1 – Supported Tape Data Encryption Out pages page

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)							
1		PAGE CODE (0001h)						(LSB)
2	(MSB)							
3		PAGE LENGTH (n-3)						(LSB)
	Supported tape data encryption out page code list							
4	(MSB)							
5		Supported tape data encryption out page code (first)						(LSB)
	:							
n-1	(MSB)							
n		Supported tape data encryption out page code (last)						(LSB)

The supported tape data encryption out page code list shall contain a list of all of the pages that the device server supports for the SECURITY PROTOCOL OUT command specifying the Tape Data Encryption security protocol in ascending order.

8.5.2.4 Data Encryption Capabilities page

Table E1 shows the format of the Data Encryption Capabilities page.

Table E1 – Data Encryption Capabilities page

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)							
1		PAGE CODE (0010h)						(LSB)
2	(MSB)							
3		PAGE LENGTH (n-3)						(LSB)
4								
19		Reserved						
	Encryption algorithm descriptor list							
20								
		Data Encryption Algorithm descriptor (first)						
	:							
n		Data Encryption Algorithm descriptor (last)						

See SPC-4 for a description of the PAGE LENGTH field.

Each Data Encryption Algorithm descriptor (see table E2) contains information about a data encryption algorithm supported by the device server. If more than one descriptor is included, they shall be sorted in ascending order of the value in the ALGORITHM INDEX field.

Table E2 – Data Encryption Algorithm descriptor

Bit / Byte	7	6	5	4	3	2	1	0
0	ALGORITHM INDEX							
1	Reserved							
2	(MSB)	DESCRIPTOR LENGTH (20)						
3								(LSB)
4	Reserved		MAC_C	DED_C	DECRYPT_C		ENCRYPT_C	
5	Reserved		NONCE_C		IV_RN	IV_EBU	IV_WPU	IV_MU
6	(MSB)	MAXIMUM UNAUTHENTICATED KEY-ASSOCIATED DATA BYTES						
7								(LSB)
8	(MSB)	MAXIMUM AUTHENTICATED KEY-ASSOCIATED DATA BYTES						
9								(LSB)
10	(MSB)	KEY SIZE						
11								(LSB)
12		Reserved						
19								
20	(MSB)	ENCRYPTION ALGORITHM IDENTIFIER						
23								(LSB)

The ALGORITHM INDEX field is a device server assigned value associated with the algorithm that is being described. The value in the ALGORITHM INDEX field is used by the SECURITY PROTOCOL OUT command Set Data Encryption page to select this algorithm.

The ENCRYPT_C field (see table E3) indicates the encryption capabilities of the device.

Table E3 - ENCRYPT_C field values

Code	Description
0	The device server has no data encryption capability using this algorithm.
1	The device server has the ability to encrypt data using this algorithm in software.
2	The device server has the ability to encrypt data using this algorithm in hardware.
3	Reserved

The DECRYPT_C field (see table E4) indicates the decryption capabilities of the device.

Table E4 - DECRYPT_C field values

Code	Description
0	The device server has no data decryption capability using this algorithm.
1	The device server has the ability to decrypt data using this algorithm in software.
2	The device server has the ability to decrypt data using this algorithm in hardware.
3	Reserved

The distinguish encrypted data capable (DED_C) bit shall be set to one if the device server is capable of distinguishing encrypted data from unencrypted data when reading it from the medium. The DED_C bit shall be set to zero if the device server is not capable of distinguishing encrypted data from unencrypted data when reading it from the medium. If the ability to distinguish encrypted data from unencrypted data is format specific and a volume is mounted, the DED_C bit shall be set based on the current format of the medium. If no volume is mounted, the DED_C bit shall be set to one if the device server is capable of distinguishing encrypted data from unencrypted data in any format that the device server supports.

The message authentication code capable (MAC_C) bit shall be set to one if the algorithm includes a message authentication code added to encrypted blocks. The MAC_C bit shall be set to zero if the algorithm does not include a message authentication code added to encrypted blocks. If the inclusion of a message authentication code is format specific and a volume is mounted, the MAC_C bit shall be set based on the current format of the medium. If no volume is mounted, the MAC_C bit shall be set to one if the device server adds a message authentication code to data encrypted with this algorithm in any format that the device server supports.

The initialization vector medium unique (IV_MU) bit shall be set to one if the initialization vector used by the encryption algorithm is unique for each medium. The IV_MU) bit shall be set to zero if the initialization vector used by the encryption algorithm is not unique for each medium.

The initialization vector write pass unique (IV_WPU) bit shall be set to one if the initialization vector used by the encryption algorithm is unique for each write operation that over writes the same portion of the medium. The IV_WPU bit shall be set to zero if the initialization vector used by the encryption algorithm is not unique for each write operation that over writes the same portion of the medium.

The initialization vector encrypted block unique (IV_EBU) bit shall be set to one if the initialization vector used by the encryption algorithm is unique for each encrypted block on the medium. The IV_EBU bit shall be set to zero if the initialization vector used by the encryption algorithm is not unique for each encrypted block on the medium.

The initialization vector random number (IV_RN) bit shall be set to one if the initialization vector used by the encryption algorithm is either in part or wholly a random number. The IV_RN bit shall be set to zero if the initialization vector used by the encryption algorithm is not in part or wholly a random number.

Table E5 describes the values in the NONCE_C field.

Table E5 - NONCE_C field values

Code	Description
0	This algorithm does not require a nonce value.
1	The device server generates the nonce value.
2	The device server requires all or part of the nonce value to be provided by the application client.
3	The device server supports all or part of the nonce value provided by the application client. If the Set Data Encryption page that enables encryption does not include a nonce value descriptor, the device server generates the nonce value.

The MAXIMUM UNAUTHENTICATED KEY-ASSOCIATED DATA BYTES field indicates the maximum size of the unauthenticated key-associated data (see 4.2.19.10) that the device server can support for this algorithm.

The MAXIMUM AUTHENTICATED KEY-ASSOCIATED DATA BYTES field indicates the maximum size of the authenticated key-associated data (see 4.2.19.10) that the device server can support for this algorithm.

The KEY SIZE field indicates the size in bytes of the encryption key required by the algorithm.

The ENCRYPTION ALGORITHM IDENTIFIER field contains an Encryption Algorithm Identifier (see SPC-4).

8.5.2.5 Supported Key Formats page

Table F1 shows the format of the Supported Key Formats page.

Table F1 – Supported Key Formats page

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)			PAGE CODE (0011h)				
1								(LSB)
2	(MSB)			PAGE LENGTH (n-3)				
3								(LSB)
4				SUPPORTED KEY FORMATS LIST				
n								

The SUPPORTED KEY FORMATS LIST field shall contain a list of all of the key formats (see table Y5) that the device server supports for the Set Data Encryption page (see 8.5.3.2) in ascending order.

8.5.2.6 Data Encryption Management Capabilities page

Table M1 shows the format of the Data Encryption Management Capabilities page.

Table M1 – Data Encryption Management Capabilities page

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)			PAGE CODE (0012h)				
1								(LSB)
2	(MSB)			PAGE LENGTH (11)				
3								(LSB)
4	Reserved							LOCK_C
5	Reserved					CKOD_C	CKORSC_C	CKORL_C
6	Reserved							
7	Reserved				AITN_C	RG_C	LOCAL_C	PUBLIC_C
8	Reserved							
15								

The LOCK_C bit shall be set to one if the device server supports the LOCK bit in the Set Data Encryption page. The LOCK_C bit shall be set to zero if the device server does not support the LOCK bit in the Set Data Encryption page.

The CKOD_C bit shall be set to one if the device server supports the CKOD bit in the Set Data Encryption page. The CKOD_C bit shall be set to zero if the device server does not support the CKOD bit in the Set Data Encryption page.

The CKORSCL_C bit shall be set to one if the device server supports the CKORSC bit in the Set Data Encryption page. The CKORSC_C bit shall be set to zero if the device server does not support the CKORSC bit in the Set Data Encryption page.

The CKORL_C bit shall be set to one if the device server supports the CKORL bit in the Set Data Encryption page. The CKORL_C bit shall be set to zero if the device server does not support the CKORL bit in the Set Data Encryption page.

The AITN_C bit shall be set to one if the device server supports a scope of ALL I_T NEXUS. The AITN_C bit shall be set to zero if the device server does not support a scope of ALL I_T NEXUS.

The RG_C bit shall be set to one if the device server supports a scope of RESERVATION GROUP. The RG_C bit shall be set to zero if the device server does not support a scope of RESERVATION GROUP.

The LOCAL_C bit shall be set to one if the device server supports a scope of LOCAL. The LOCAL_C bit shall be set to zero if the device server does not support a scope of LOCAL.

The PUBLIC_C bit shall be set to one if the device server supports a scope of PUBLIC. The PUBLIC_C bit shall be set to zero if the device server does not support a scope of PUBLIC.

8.5.2.7 Data Encryption Status page

Table S1 shows the format of the Data Encryption Status page.

Table S1 – Data Encryption Status page

Byte \ Bit	7	6	5	4	3	2	1	0
0	(MSB)			PAGE CODE (0020h)				
1								(LSB)
2	(MSB)			PAGE LENGTH (n-3)				
3								(LSB)
4	I_T NEXUS SCOPE			Reserved		KEY SCOPE		
5	ENCRYPTION MODE							
6	DECRYPTION MODE							
7	ALGORITHM INDEX							
8	(MSB)			KEY INSTANCE COUNTER				
11								(LSB)
12	(MSB)			Reserved				
23								(LSB)
24				KEY-ASSOCIATED DATA DESCRIPTORS LIST				
n								

The I_T NEXUS SCOPE field shall contain the value from the data encryption scope saved for the I_T nexus on which this command was received (see 4.2.19.6).

The KEY SCOPE field shall contain the value from the key scope in the saved data encryption parameters currently associated with the I_T nexus on which this command was received (see 4.2.19.7).

The ENCRYPTION MODE field shall contain the value from the encryption mode in the saved data encryption parameters currently associated with the I_T nexus on which this command was received (see 4.2.19.7).

The DECRYPTION MODE field shall contain the value from the decryption mode in the saved data encryption parameters currently associated with the I_T nexus on which this command was received (see 4.2.19.7).

The ALGORITHM INDEX field shall contain the value from the algorithm index in the saved data encryption parameters currently associated with the I_T nexus on which this command was received (see 4.2.19.7). If the ENCRYPTION MODE field and the DECRYPTION MODE field are both set to DISABLE, the value in the ALGORITHM INDEX field is undefined.

The KEY INSTANCE COUNTER field contains the value of the key instance counter (see 4.2.19.8) assigned to the key indicated by the KEY SCOPE field value.

If the ENCRYPTION MODE field and the DECRYPTION MODE field are both set to DISABLE, the KEY-ASSOCIATED DATA DESCRIPTORS LIST field shall not be included in the page.

If either the ENCRYPTION MODE field or the DECRYPTION MODE field are set to a value other than DISABLE, the KEY-ASSOCIATED DATA DESCRIPTORS LIST field shall contain data security descriptors (see 8.5) describing attributes assigned to the key defined by the I_T NEXUS SCOPE and KEY SCOPE fields at the time the key was established in the device server by processing a Set Data Encryption page. If more than one key associated descriptor is included, they shall be order of increasing value of the DESCRIPTOR TYPE field. Descriptors shall be included as defined by the following paragraphs.

An unauthenticated key-associated data descriptor (see 8.5.4.3) shall be included if an unauthenticated key-associated data descriptor was included in the Set Data Encryption page that established the key in the device server. The AUTHENTICATED field shall be set to zero. The KEY DESCRIPTOR field shall contain the U-KAD value associated with the key.

An authenticated key-associated data descriptor (see 8.5.4.4) shall be included if an authenticated key-associated data descriptor was included in the Set Data Encryption page that established the key in the device server. The AUTHENTICATED field shall be set to zero. The KEY DESCRIPTOR field shall contain the A-KAD value associated with the key.

A nonce value descriptor (see 8.5.4.5) shall be included if a nonce value descriptor was included in the Set Data Encryption page that established the key in the device server. The AUTHENTICATED field shall be set to zero. The KEY DESCRIPTOR field shall contain the nonce value associated with the key. A nonce value descriptor may be included if no nonce value descriptor was included in the Set Data Encryption page that established the key in the device server. In this case, the KEY DESCRIPTOR field shall be set to the nonce value established by the device server for use with the selected key.

8.5.2.8 Next Block Encryption Status page

Table N1 shows the format of the Next Block Encryption Status page.

Table N1 – Next Block Encryption Status page

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)							
1				PAGE CODE (0021h)				(LSB)
2	(MSB)							
3				PAGE LENGTH (n-3)				(LSB)
4	(MSB)							
11				LOGICAL OBJECT NUMBER				(LSB)
12	COMPRESSION STATUS				ENCRYPTION STATUS			
13	ALGORITHM INDEX							
14								
15				Reserved				
16								
n				KEY-ASSOCIATED DATA DESCRIPTORS				

The LOGICAL OBJECT NUMBER field contains the logical object identifier of the next logical object (see 4.2.5.2)

The COMPRESSION STATUS field is described in table N2.

Table N2 – COMPRESSION STATUS field values

Code	Description
0h	The device server is incapable of determining if the logical object referenced by the LOGICAL OBJECT NUMBER field has been compressed.
1h	The device server is capable of determining if the logical object referenced by the LOGICAL OBJECT NUMBER field has been compressed, but is not able to at this time. Possible reasons are: a) The next logical block has not yet been read into the buffer; b) There was an error reading the next logical block; or c) There are no more logical blocks (i.e., end of data).
2h	The device server has determined that the logical object referenced by the LOGICAL OBJECT NUMBER field is not a logical block
3h	The device server has determined that the logical object referenced by the LOGICAL OBJECT NUMBER field is not compressed.
4h	The device server has determined that the logical object referenced by the LOGICAL OBJECT NUMBER field is compressed.
5h – Fh	Reserved

The ENCRYPTION STATUS field is described in table N3.

Table N3 – ENCRYPTION STATUS field values

Code	Description
0h	The device server is incapable of determining if the logical object referenced by the LOGICAL OBJECT NUMBER field has been encrypted.
1h	The device server is capable of determining if the logical object referenced by the LOGICAL OBJECT NUMBER field has been encrypted, but is not able to at this time. Possible reasons are: a) The next logical block has not yet been read into the buffer; b) There was an error reading the next logical block; or c) There are no more logical blocks (i.e: end of data).
2h	The device server has determined that the logical object referenced by the LOGICAL OBJECT NUMBER field is not a logical block
3h	The device server has determined that the logical object referenced by the LOGICAL OBJECT NUMBER field is not encrypted.
4h	The device server has determined that the logical object referenced by the LOGICAL OBJECT NUMBER field is encrypted by an algorithm that is not supported by this device server. The values in the KEY-ASSOCIATED DATA DESCRIPTORS fields contain information pertaining to the encrypted block.
5h	The device server has determined that the logical object referenced by the LOGICAL OBJECT NUMBER field is encrypted by an algorithm supported by this device server. The values in the ALGORITHM INDEX and KEY-ASSOCIATED DATA DESCRIPTORS fields contain information pertaining to the encrypted block.
6h	The device server has determined that the logical object referenced by the LOGICAL OBJECT NUMBER field is encrypted by an algorithm supported by this device server but the device server either is not enabled to decrypt or does not have the correct key or nonce value to decrypt the data.
7h – Fh	Reserved

The ALGORITHM INDEX field indicates which of the encryption algorithms reported by the SECURITY PROTOCOL IN command Data Encryption Capabilities page was used to encrypt the logical block. For values in the ENCRYPTION STATUS field (see table N3) that do not indicate the ALGORITHM INDEX field is valid, the algorithm index is undefined.

If the encryption status indicates that the next logical block is encrypted by a supported algorithm, the device server shall include in the KEY-ASSOCIATED DATA DESCRIPTORS field all key-associated data that is associated with the encrypted block that was recorded on the medium. If more than one key-associated data descriptor is include in the Next Block Encryption Status page, they shall be in increasing numeric order of the value in the DESCRIPTOR TYPE field.

An unauthenticated key-associated data descriptor (see 8.5.4.3) shall be included if any unauthenticated key-associated data is associated with the next logical block. The AUTHENTICATED field shall be set to zero. The KEY DESCRIPTOR field shall contain the U-KAD value associated with the encrypted block.

An authenticated key-associated data descriptor (see 8.5.4.4) shall be included if any authenticated key-associated data is associated with the next logical block. The AUTHENTICATED field shall indicate the status of the authentication done by the device server. The KEY DESCRIPTOR field shall contain the A-KAD value associated with the encrypted block.

A nonce value descriptor (see 8.5.4.5) shall be included if a nonce value was not generated by the device server (i.e., it was established by a nonce value descriptor included in the Set Data Encryption page use to set the key and algorithm used to encrypt the logical block.) or if the device server can not determine if the nonce was generated by the device server that encrypted the logical block. A nonce value descriptor may be included if the nonce value was generated by the device server that encrypted the logical block. The AUTHENTICATED field shall indicate the status of the authentication done by the device server. The KEY DESCRIPTOR field shall contain the nonce value associated with the encrypted block

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

The SECURITY PROTOCOL OUT command specifying the Tape Data Encryption security protocol (i.e., 20h) is used to configure the data security methods in the device server and on the medium. The command supports a series of pages that are requested individually. An application client requests a page by using a SECURITY PROTOCOL OUT command with the SECURITY PROTOCOL field set to Tape Data Encryption security protocol and the SP_SPECIFIC field set to the page code requested.

The SP_SPECIFIC field (see Table B2) indicates the type of page that the application client is sending.

Table B2 – SP_SPECIFIC field values

Code	Description	Reference
00h – 0Fh	Reserved	
10h	Set Data Encryption page	8.5.3.2
11h - FFh	Reserved	

If the SP_SPECIFIC field is set to a reserved or unsupported value, 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 CDB.

8.5.3.2 Set Data Encryption page

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

Table Y1 – Set Data Encryption page

Byte \ Bit	7	6	5	4	3	2	1	0
0	(MSB)							
1				PAGE CODE (0010h)				(LSB)
2	(MSB)							
3				PAGE LENGTH (m-3)				(LSB)
4	SCOPE				Reserved			LOCK
5	Reserved					CKOD	CKORSC	CKORL
6	ENCRYPTION MODE							
7	DECRYPTION MODE							
8	ALGORITHM INDEX							
9	KEY FORMAT							
10				Reserved				
17								
18	(MSB)							
19				KEY LENGTH (n-19)				(LSB)
20				KEY				
n								
n+1				KEY-ASSOCIATED DATA DESCRIPTORS LIST				
m								

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 SCOPE field (see table Y2) indicates the scope of the data encryption parameters. Support for scope values of PUBLIC and ALL I_T NEXUS are mandatory for device servers that support the Set Data Encryption page.

Table Y2 – SCOPE field values

Code	Name	Description
0	PUBLIC	All fields other than the SCOPE field and lock bit shall be ignored. The I_T nexus shall use data encryption parameters that are shared by other I_T nexuses. If no I_T nexuses are sharing data encryption parameters, the device server shall use default data encryption parameters.
1	LOCAL	The data encryption parameters are unique to the I_T nexus associated with the SECURITY PROTOCOL OUT command and shall not be shared with other I_T nexuses.
2	RESERVATION GROUP	The data encryption parameters shall be shared with all participants in a reservation.
3	ALL I_T NEXUS	The data encryption parameters shall be shared with all I_T nexuses.
4 – 7		Reserved

See 4.2.19.9 for a description of the LOCK bit.

If the clear key on de-mount (CKOD) bit is set to one the device server shall set the data encryption parameters to default values upon completion of a volume de-mount. If the CKOD bit is set to zero, the de-mounting of a volume shall not affect the data encryption parameters. If the CKOD bit is set to one and there is no volume mounted in the device, 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 DATA.

If the clear key on reservation scope change (CKORSC) bit is set to one the device server shall set the data encryption parameters to default values on the change in scope of the persistent reservation. If the CKORSC bit is set to zero, a change in scope of a persistent reservation shall not affect the data encryption parameters. If the CKORSC bit is set to one and there is no persistent reservation in effect for the I_T nexus associated with the SECURITY PROTOCOL OUT command, 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 DATA.

If the clear key on reservation loss (CKORL) bit is set to one the device server shall set the data encryption parameters to default values on a reservation loss (see 3.1.B). If the CKORL bit is set to zero, a reservation loss shall not affect the data encryption parameters. If the CKORL bit is set to one and there is no reservation in effect for the I_T nexus associated with the SECURITY PROTOCOL OUT command, 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 DATA.

Table Y3 defined the values for the ENCRYPTION MODE field.

Table Y3 – ENCRYPTION MODE field values

Code	Name	Description
0h	DISABLE	Data encryption is disabled.
1h	EXTERNAL	The data associated with WRITE(6) and WRITE(16) commands has been encrypted by a system that is compatible with the algorithm specified by the ALGORITHM INDEX field.
2h	ENCRYPT	The device server shall encrypt all data that it receives for a WRITE(6) or WRITE(16) using the algorithm specified in the ALGORITHM INDEX field and the key provided in the KEY field.
3h – Fh		Reserved

Table Y4 defined the values for the DECRYPTION MODE field. See 4.2.19.3 for configuration and exception condition requirements.

Table Y4 – DECRYPTION MODE field values

Code	Name	Description
0h	DISABLE	Data decryption is disabled. If the device server encounters an encrypted block while reading, it shall not allow access to the data.
1h	RAW	Data decryption is disabled. If the device server encounters an encrypted block while reading, it shall pass the encrypted block to the host without decrypting it. The encrypted block may contain data that is not user data.
2h	DECRYPT	The device server shall decrypt all data that is read from the medium in response to a READ(6), READ(16), READ REVERSE(6), READ REVERSE(16), or RECOVER BUFFERED DATA command or verified when processing a VERIFY(6) or VERIFY(16) command. The data shall be decrypted using the algorithm specified in the ALGORITHM INDEX field and the key provided in the KEY field.
3h	MIXED	The device server shall decrypt all data that is read from the medium that it determines was encrypted in response to a READ(6), READ(16), READ REVERSE(6), READ REVERSE(16), or RECOVER BUFFERED DATA command or verifying when processing a VERIFY(6) or VERIFY(16) command. The data shall be decrypted using the algorithm specified in the ALGORITHM INDEX field and the key provided in the KEY field. If the device server encounters unencrypted data when processing a READ(6), READ(16), READ REVERSE(6), READ REVERSE(16), RECOVER BUFFERED DATA, VERIFY(6), or VERIFY(16) command, the data shall be processed without decrypting.
4h – Fh		Reserved

If the device server does is not capable of distinguishing encrypted blocks from unencrypted blocks using the algorithm specified in the ALGORITHM INDEX field and the DECRYPTION MODE field is set to MIXED, 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 DATA.

If the ENCRYPTION MODE field is set to ENCRYPT and the KEY LENGTH field is set to zero, 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 DATA.

If the DECRYPTION MODE field is set to DECRYPT or MIXED and the KEY LENGTH field is set to zero, 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 DATA.

The ALGORITHM INDEX field indicates which of the encryption algorithms reported by the SECURITY PROTOCOL IN command Data Encryption Capabilities pages shall be used to encrypt and decrypt data.

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

Table Y5 – KEY FORMAT field values

Code	Description
00h	The KEY field contains the key to be used to encrypt or decrypt data.
01h	The KEY field contains a vendor specific key reference
02h – BFh	Reserved
C0h – FFh	Vendor specific

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 Y8 – KEY field contents with KEY FORMAT field set to 00h

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)	key						
n								(LSB)

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

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 ENCRYPTION MODE field is set to ENCRYPT the device server shall save the key-associated descriptors in the KEY-ASSOCIATED DATA DESCRIPTORS LIST field and associate them with every logical block that is encrypted with this key by the device server. If more than one key-associated data descriptor is include in the Set Data Encryption page, they shall be in increasing numeric order of the value in the DESCRIPTOR TYPE field. If the ENCRYPTION MODE field is not set to ENCRYPT and key-associated descriptors are included in the KEY-ASSOCIATED DATA DESCRIPTORS LIST field, the device server shall terminate the command with CHECK CONDITION, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN PARAMETER LIST.

An unauthenticated key-associated data descriptor (see 8.5.4.3) may be included if any unauthenticated key-associated data is to be associated with logical blocks encrypted with the algorithm and key. The AUTHENTICATED field shall be set to zero. The KEY DESCRIPTOR field shall contain the U-KAD value associated with the encrypted block.

An authenticated key-associated data descriptor (see 8.5.4.4) may be included if any authenticated key-associated data is to be associated with logical blocks encrypted with the algorithm and key. The AUTHENTICATED field shall be set to zero. The KEY DESCRIPTOR field shall contain the A-KAD value associated with the encrypted block.

If a nonce value descriptor (see 8.5.4.5) is included and the algorithm and the device server supports application client generated nonce values, the value in the KEY DESCRIPTOR field shall be used as the nonce value for the encryption process. If a nonce value descriptor is included and the encryption algorithm or the device server does not support application client generated nonce values, the device server shall terminate the command with CHECK CONDITION, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN PARAMETER LIST. If the encryption algorithm or the device server requires an application client generated nonce value and a nonce value descriptor is not included, the device server shall terminate the command with CHECK CONDITION, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INCOMPLETE KEY-ASSOCIATE DATA SET. If a nonce value descriptor is included, the AUTHENTICATED field shall be set to zero. The KEY DESCRIPTOR field shall contain the nonce value associated with the encrypted block

8.5.4 SECURITY PROTOCOL IN and SECURITY PROTOCOL OUT descriptors

8.5.4.1 Tape Data Encryption security protocol descriptors overview

Several of the parameter pages in used by the SECURITY PROTOCOL IN and SECURITY PROTOCOL OUT commands allow for the inclusion of descriptors to provide additional optional data. This subclause defines the descriptors that are common between multiple pages.

8.5.4.2 Tape Data Encryption descriptors format

Each Tape Data Encryption descriptor shall take the form as defined in table D1.

Table D1 –Tape Data Encryption descriptor format

Bit / Byte	7	6	5	4	3	2	1	0
0	KEY DESCRIPTOR TYPE							
1	Reserved						AUTHENTICATED	
2	(MSB)	KEY DESCRIPTOR LENGTH (n-3)						
3								(LSB)
4	(MSB)	KEY DESCRIPTOR						
n								(LSB)

The KEY DESCRIPTOR TYPE field contains a value from table D2 that defines the contents of the KEY DESCRIPTOR field.

Table D2 – KEY DESCRIPTOR TYPE field values

Code	Description	Reference
00h	Unauthenticated key-associated data	8.5.4.3
01h	Authenticated key-associated data	8.5.4.4
02h	Nonce value	8.5.4.5
03h – BFh	Reserved	
C0h – FFh	Vendor specific	

Table D3 defines the values for the AUTHENTICATED field.

Table D3 – AUTHENTICATED field values

Code	Description
0	The value in the KEY DESCRIPTOR field is not covered by the authentication (i.e. U-KAD)
1	No attempt has been made to authenticate the value in the KEY DESCRIPTOR field.
2	The value in the KEY DESCRIPTOR field has been authenticated.
3	The value in the KEY DESCRIPTOR field has failed authentication.
4 - 7	Reserved

8.5.4.3 Unauthenticated key-associated data key descriptor

The AUTHENTICATED field in an unauthenticated key-associated data descriptor shall be set to zero.

The KEY DESCRIPTOR field of an unauthenticated key-associated data descriptor shall contain any unauthenticated key-associated data assigned to the key.

8.5.4.4 Authenticated key-associated data key descriptor

The AUTHENTICATED field shall be set as defined by the page in which the descriptor is included.

The KEY DESCRIPTOR field of an authenticated key-associated data descriptor shall contain any authenticated key-associated data assigned to the key.

8.5.4.5 Nonce value descriptor

The AUTHENTICATED field shall be set as defined by the page in which the descriptor is included.

The KEY DESCRIPTOR field of a nonce value descriptor shall contain the nonce value used with the data encryption key.