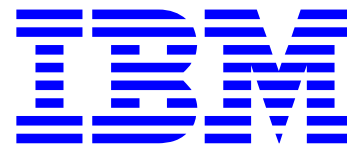


To: INCITS Technical Committee T10  
From: Kevin Butt  
Date: Monday, October 06, 2008 6:53 pm  
Document: T10/08-391r0 — SSC-3: Resolution to LB IBM-L2



## Revisions

08-391r0 (06 October 2008) Initial revision. I do not believe this has completely resolved the issue yet, but there have been many edits and it is worth reviewing in the working group.

## Introduction

This proposal intends to resolve LB IBM-L2:

[4.2.21.11, p2] Add a new sentence after s1:

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. A set of data encryption parameters are established and locked even if the ENCRYPTION MODE is set to DISABLE and the DECRYPTION MODE is set to DISABLE.

Proposal 08-266r0 (<http://www.t10.org/ftp/t10/document.08/08-266r0.ppt>) was reviewed in the past and the working group confirmed the LOCK bit is supposed to work as shown in that document. 08-266r0 was just a presentation to gain consensus and not a proposal. It is my intent to make a proposal that clarifies the intended usage of the LOCK bit as the usage is shown in 08-266r0. Unfortunately, 08-266r0 does not discuss the LOCK bit in relation to PUBLIC and encryption modes set to DISABLE.

When this issue was first raised there were a series of emails that went back and forth between Paul Entzel and me. This can be found in the reflector archives at [http://www.t10.org/ftp/t10/t10r/2008/r0804021\\_f.htm](http://www.t10.org/ftp/t10/t10r/2008/r0804021_f.htm)

On further review of SSC-3 I find that Table 142 - SCOPE field values has as part of the description for PUBLIC the following: "All fields other than the SCOPE field and LOCK bit shall be ignored."

**The discussion of this topic, as well as the writing of this proposal, has been very confusing. This proposal needs close scrutiny to make sure it meets the intent of the LOCK bit and the data encryption scope.** Paul Entzel believes the proposed solution in the actual letter ballot comment to be incorrect. I agree it is, at least incomplete. The interactions between the LOCK bit, the data encryption parameters and the saved information per I\_T nexus are not clear and need to be clarified. These issues are all inter-related to the issue that drove the LB comment about the LOCK bit.

## Proposal

### 4.2.21 Data encryption

#### 4.2.21.1 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 8.5.2 on 172 and 8.5.3 on 189) 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.

#### 4.2.21.2 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\_L nexus basis. The data encryption process is enabled for an I\_T\_L nexus upon successful completion of a SECURITY PROTOCOL OUT command that sends a Set Data Encryption page (see 8.5.3.2 on page 190) with the ENCRYPTION MODE field set to ENCRYPT and with a valid key. If the data encryption scope parameter for an I\_T\_L nexus is set to PUBLIC (see 4.2.21.7), the data encryption process may be enabled by another I\_T\_L nexus that establishes a set of data encryption parameters with a **key-scope data encryption scope** of ALL I\_T NEXUS (see 4.2.21.8).

If data encryption is enabled for an I\_T\_L nexus and the mounted volume supports the selected encryption algorithm at the current logical position, all logical blocks received by the device server from that I\_T\_L nexus as part of a WRITE(6) or WRITE(16) command shall be encrypted before being recorded on the medium. Filemarks are logical objects that shall not be encrypted.

If data encryption is enabled for an I\_T\_L nexus and the mounted volume does not support the selected encryption algorithm at the current logical position, then the device server shall terminate a WRITE(6) or WRITE(16) command with CHECK CONDITION status, with the sense key set to DATA PROTECT, and the additional sense code set to ENCRYPTION PARAMETERS NOT USEABLE.

If data encryption is enabled for an I\_T\_L nexus and the mounted volume does not support the selected encryption algorithm at the current logical position, then the device server may terminate a WRITE FILEMARKS(6) or WRITE FILEMARKS(16) command with CHECK CONDITION status, with the sense key set to DATA PROTECT, and the additional sense code set to ENCRYPTION PARAMETERS NOT USEABLE.

#### 4.2.21.3 Reading encrypted blocks on the medium

---

---

Editors Note 1 - KDB: No changes in this section

---

---

#### 4.2.21.4 Exhaustive-search attack prevention

---

---

Editors Note 2 - KDB: No changes in this section

---

---

#### 4.2.21.5 Keyless copy of encrypted data

---

---

Editors Note 3 - KDB: No changes in this section

---

---

#### 4.2.21.6 Managing keys within the physical device

To increase the security of keys, the data encryption parameters are volatile in the physical device and the data encryption keys are never reported to an application client. The physical device 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 147).

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 147*) in the SUPPORTED KEY FORMATS LIST field in the Supported Key Formats page (see 8.5.2.5).

The physical device shall release the resources used to save a set of data encryption parameters (see 4.2.21.8) under the following conditions:

- a) the CKOD bit is set to one in the saved data encryption parameters and the volume is demounted;
- b) the CKORL bit is set to one and the ~~key-scope~~ data encryption 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~~ data encryption scope is set to ALL I\_T NEXUS in the saved data encryption parameters and the device server experiences a reservation loss (see 3.1.56 on page 7);
- d) the CKORP bit is set to one in the saved data encryption parameters and the device server processes a PERSISTENT RESERVE OUT command with a service action of either PREEMPT or PREEMPT AND ABORT;
- e) a microcode update is performed on the device; or
- f) a power on condition occurs.

The physical device may release the resources used to save a set of data encryption parameters if:

- a) a volume is mounted that does not support data encryption using the algorithm specified by the algorithm index in the data encryption parameter; or
- b) other vendor-specific events.

---



---

Editors Note 4 - DAP: data encryption parameter is ambiguous to me, need to specify the page/descriptor.

---



---

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 physical device 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 I\_T NEXUS for all other I\_T nexus that has its registered for encryption unit attentions state set to one (see 4.2.21.7) and is 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 keys~~).

If a device server processes a Set Data Encryption page that includes a key and the SDK bit is set to zero, the device server shall establish a unit attention condition with the additional sense code set to DATA ENCRYPTION PARAMETERS CHANGED BY ANOTHER I\_T NEXUS for all other I\_T nexus that have their registered for encryption unit attentions state set to one (see 4.2.21.7) and 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~~), at and the physical device 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 set of data encryption parameters with the values from the Set Data Encryption page.

A physical device shall save at most one set of data encryption parameters with a ~~key-scope~~ data encryption 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 establish a unit attention condition with the additional sense code set to DATA ENCRYPTION PARAMETERS CHANGED BY ANOTHER I\_T NEXUS for all other I\_T nexus that have their registered for encryption unit attentions state set to one (see 4.2.21.7) and 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~~) and the physical device shall:

- a) release any resources that it had allocated to store data encryption parameters with a ~~key-scope~~ data encryption 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 set of data encryption parameters with the values from the Set Data Encryption page and a ~~key-scope~~ data encryption scope value of ALL I\_T NEXUS.

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 has its registered for encryption unit attentions state set to one (see 4.2.21.7) and is affected by the change of the key.

#### 4.2.21.7 ~~Saved information per I\_T nexus~~ data encryption information

---



---

Editors Note 5 - KDB: It seems that this section should focus on I\_T nexus data encryption information. What that information is and when each element is set to what. There should be a different clause stating the what and when for data encryption parameters. These two sections should not overlap or specify stuff that belongs in the other like they currently seem to do.

---



---

If the device server supports data encryption it shall maintain I T nexus data encryption information on a per I T nexus basis. This I T nexus data encryption information shall contain: ~~save the following information on a per I\_T nexus basis:~~

- a) data encryption scope for this I T nexus;
- b) lock value for this I T nexus;
- c) key instance counter value at ~~lock~~ the time lock was set to one for this I T nexus;
- 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 and the SDK bit ~~is~~ set to zero; and
- e) registered for encryption unit attentions state for this I T nexus.

The set of possible data encryption scope values for an I\_T nexus is:

- a) PUBLIC;
- b) LOCAL; or
- c) ALL I\_T NEXUS

---



---

Editors Note 6 - KDB: This is the second place this is defined. Can we reference one to the other? Or is this intended to say, "The set of possible values for the data encryption scope contained in the I\_T nexus data encryption information is:" in which case the statement below about supporting all values supported in the data encryption parameters seems to cover this and this paragraph is not needed.

---



---

If ~~an I\_T nexus data encryption scope~~ the data encryption scope contained in the I T nexus data encryption information for this I T nexus is set to PUBLIC it indicates the physical device does not have a saved set of data encryption parameters that were established by ~~that this I\_T nexus.~~ ~~Device servers that support encryption shall support an I\_T nexus data encryption scope of PUBLIC.~~

---



---

Editors Note 7 - KDB: The last sentence above does not belong in this section. It belongs in the data

encryption parameters section. I moved it there

---



---

Editors Note 8 - KDB: Should the data encryption parameters (not the saved information per I\_T nexus) be set to default values when the data encryption scope is set to PUBLIC? If yes, what are the default values?

---



---

When a device server successfully completes the processing of a Set Data Encryption page with a scope of PUBLIC through this I\_T nexus the device server shall:

- a) set the data encryption scope contained in the I\_T nexus data encryption information for this I\_T nexus to PUBLIC;
  - b) set the lock contained in the I\_T nexus data encryption information for this I\_T nexus to the value of the LOCK bit in the Set Data Encryption page;
- 
- 

Editors Note 9 - KDB: I got the impression that Paul Entzel did not agree with item b when we had our discussion on the reflector. However, if Table 142 is correct, then this has to be the behavior.

---



---

- c) set the key instance counter value at the time lock was set to one contained in the I\_T nexus data encryption information for this I\_T nexus to zero;
- d) set the 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 and the SDK bit set to zero contained in the I\_T nexus data encryption information for this I\_T nexus to zero; and
- e) set the registered for encryption unit attentions state contained in the I\_T nexus data encryption information for this I\_T nexus to zero.

A device server shall set the data encryption scope contained in the I\_T nexus data encryption information for an this I\_T nexus to LOCAL when it successfully completes the processing of a Set Data Encryption page with a scope of LOCAL ~~from that through this 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 through~~ the I\_T nexus that was used to established the parameters. ~~A physical device 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. If the physical device releases resources used to store a set of data encryption parameters with a data encryption scope of LOCAL, 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 contained in the I\_T nexus data encryption information for an this 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 through this I\_T nexus~~. At most, one I\_T nexus shall be assigned the data encryption scope of ALL I\_T NEXUS. If the physical device releases resources used to store a set of data encryption parameters with a ~~key scope~~ data encryption 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.~~

Device servers shall support setting the data encryption scope contained in the I\_T nexus data encryption information to any value of data encryption scope supported in the data encryption parameters.

~~By default, the device server shall set the saved I\_T nexus parameters data encryption scope value to PUBLIC and lock value to zero. Table 1 defines the values assigned the I\_T nexus data encryption information when a power on condition occurs.~~

**Table {new}1 — Default I\_T nexus data encryption information**

<u>Parameter</u>	<u>Value when a power on condition occurs</u>
<u>data encryption scope</u>	<u>PUBLIC</u>
<u>lock</u>	<u>zero</u>
<u>key instance counter value at the time lock was set to one</u>	<u>zero</u>
<u>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 and the SDK bit set to zero</u>	<u>zero</u>
<u>registered for encryption unit attentions state</u>	<u>zero</u>

~~A device server shall set the lock contained in the I\_T nexus data encryption information for this I\_T nexus to the value of the LOCK bit in the Set Data Encryption page when it successfully completes the processing of a Set Data Encryption page through this I\_T nexus.~~

~~A device server shall set the key instance counter value contained in the I\_T nexus data encryption information for this I\_T nexus to the value of the key instance counter in the data encryption parameters when it successfully completes the processing of a Set Data Encryption page through this I\_T nexus and the LOCK bit in the Set Data Encryption page is set to one.~~

~~A device server shall set the 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 and the SDK bit set to zero contained in the I\_T nexus data encryption information for this I\_T nexus to the value of the key instance counter in the data encryption parameters when it successfully completes the processing of a Set Data Encryption page through this I\_T nexus and the SCOPE field is set to LOCAL and the SDK bit is set to zero in the Set Data Encryption page.~~

The registered for encryption unit attentions state contained in the I\_T nexus data encryption information is a single bit state variable that indicates if the device server shall generate unit attention conditions related to encryption status for ~~the this I\_T nexus~~. The device server shall set the registered for encryption unit attentions state contained in the I\_T nexus data encryption information to one for ~~an this I\_T nexus~~ if the device server processes a:

- a) SECURITY PROTOCOL IN command specifying the Tape Data Encryption protocol from ~~the this I\_T nexus~~; or
- b) SECURITY PROTOCOL OUT command specifying the Tape Data Encryption protocol from ~~the this I\_T nexus~~.

The device server shall set the registered for encryption unit attentions state contained in the I\_T nexus data encryption information to zero for an ~~I\_T nexus~~ if an I\_T nexus loss occurs for that I\_T nexus. The device server shall set the registered for encryption unit attentions state contained in the I\_T nexus data encryption information to zero for all ~~I\_T nexus~~ if the device server processes a logical unit reset.

#### 4.2.21.8 Data encryption parameters

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

- a) for SCSI transport protocols where SCSI initiator device port names are required, the SCSI initiator device port name; otherwise, the SCSI initiator device port identifier;
- b) indication of the SCSI target port through which the data encryption parameters were established;
- c) ~~key-scope~~ data encryption scope;
- d) encryption mode;
- e) decryption mode;
- f) key;
- g) supplemental decryption keys where supported;
- h) algorithm index;
- i) key instance counter;
- j) CKOD;
- k) CKORL;
- l) CKORP;
- m) U-KAD;
- n) A-KAD;
- o) M-KAD;
- p) nonce;
- q) raw decryption mode disable where supported; and
- r) check external encryption mode where supported.

A physical device 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 data encryption scope on every I\_T nexus that it is capable of managing). A physical device may release a previously established set of data encryption parameters when a Set Data Encryption page is processed ~~and-if~~ there are no unused resources available. The method of choosing which set of data encryption parameters to release is vendor specific. If the physical device does release a previously established set of data encryption parameters to free the resource, the device server shall establish a unit attention condition for every affected I\_T nexus (see 4.2.21.6) that has its registered for encryption unit attentions state set to one (see 4.2.21.7). A physical device is not required to have separate resources to store data encryption parameters for every data encryption scope that is supported.

A device server that supports data encryption shall support ~~an-encryption-key-a~~ data encryption scope value of ALL I\_T NEXUS and the physical device shall have resources to save one set of data encryption parameters with this data encryption scope.

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

The physical device may or may not have resources to save a set of data encryption parameters with a data encryption scope value of ALL I T NEXUS and a set of data encryption parameters with a data encryption scope value of LOCAL at the same time.

A device server that supports data encryption shall support a data encryption scope value of PUBLIC.

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 contained in the I T nexus data encryption information for the I\_T nexus is set to LOCAL or ALL I\_T NEXUS (see 4.2.21.7), the data encryption parameters set by the last Set Data Encryption page from that I\_T nexus; or

- b) if the data encryption scope contained in the I\_T nexus data encryption information for the I\_T nexus is set to PUBLIC:
  - 1) the data encryption parameters that have been saved by the physical device with a key-scope data encryption scope of ALL I\_T NEXUS if any data encryption parameters have been saved with this key-scope data encryption scope; or
  - 2) the default data encryption parameters.

#### 4.2.21.9 Data encryption capabilities

A physical device that supports data encryption shall have a set of data encryption capabilities. The set of data encryption capabilities determine the values reported through a SECURITY PROTOCOL IN command specifying the Tape Data Encryption security protocol and the Data Encryption Capabilities page (see clause 8.5.2.4). The set of data encryption capabilities includes the set of data encryption algorithms supported by the physical device.

The set of data encryption capabilities includes some values which may be changed by a method beyond the scope of this standard. The capabilities that may be changed include:

- a) the set of data encryption algorithms reported by the device server;
- b) encryption capable;
- c) decryption capable; and
- d) other vendor-specific data encryption capabilities.

#### 4.2.21.10 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, except the supplemental decryption keys, 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 the SECURITY PROTOCOL IN command. The key instance counters are 32 bits and shall roll over to zero when incremented past their maximum value.

#### 4.2.21.11 Encryption mode locking

There are conditions outside of the control of an application client which cause the physical device to release the resources used to save the data encryption parameters (see 4.2.21.6) or change the data encryption parameters used to control the encryption of logical blocks. 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 set of data encryption parameters established at the completion of the processing of the command to 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 set of data encryption parameters and key instance counter value I\_T nexus remains locked to that I\_T nexus 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~~ has a set of data encryption parameters and key instance counter locked to that I\_T nexus, and the key instance counter value has changed since the time it was locked (i.e., the key instance counter of the data encryption parameters does not equal the key instance counter of the I\_T nexus data encryption information for this I\_T nexus), 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 in this manner 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.

---

---

Editors Note 10 - DAP: Review all (red) text in this subclause with respect to data vs logical block and I\_T nexus vs I\_T\_L nexus.

---

---

---

---

Editors Note 11 - KDB: No changes in this rest of this clause.

---

---

## 8.5 Security protocol parameters

### 8.5.1 Security protocol overview

---

---

Editors Note 12 - KDB: No changes in this section

---

---

### 8.5.2 SECURITY PROTOCOL IN command specifying Tape Data Encryption security protocol

---

---

Editors Note 13 - KDB: No changes in this section

---

---

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

---

---

Editors Note 14 - KDB: No changes in this section

---

---

#### 8.5.2.2 Tape Data Encryption In Support page

---

---

Editors Note 15 - KDB: No changes in this section

---

---

#### 8.5.2.3 Tape Data Encryption Out Support page

---

---

Editors Note 16 - KDB: No changes in this section

---

---

#### 8.5.2.4 Data Encryption Capabilities page

---

---

Editors Note 17 - KDB: No changes in this section

---

---

#### 8.5.2.5 Supported Key Formats page

---

---

Editors Note 18 - KDB: No changes in this section

---

---

### 8.5.2.6 Data Encryption Management Capabilities page

Table 2 specifies the format of the Data Encryption Management Capabilities page.

**Table 2 — Data Encryption Management Capabilities page**

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB) PAGE CODE (0012h) (LSB)							
1								
2	(MSB) PAGE LENGTH (12) (LSB)							
3								
4	Reserved							LOCK_C
5	Reserved					CKOD_C	CKORP_C	CKORL_C
6	Reserved							
7	Reserved					AITN_C	LOCAL_C	PUBLIC_C
8	Reserved							
15	Reserved							

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 CKORP\_C bit shall be set to one if the device server supports the CKORP bit in the Set Data Encryption page. The CKORP\_C bit shall be set to zero if the device server does not support the CKORP 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 data encryption scope](#) of ALL I\_T NEXUS. The AITN\_C bit shall be set to zero if the device server does not support a [scope data encryption scope](#) of ALL I\_T NEXUS.

The LOCAL\_C bit shall be set to one if the device server supports a [scope data encryption scope](#) of LOCAL. The LOCAL\_C bit shall be set to zero if the device server does not support a [scope data encryption scope](#) of LOCAL.

The PUBLIC\_C bit shall be set to one if the device server supports a [scope data encryption scope](#) of PUBLIC. The PUBLIC\_C bit shall be set to zero if the device server does not support a [scope data encryption scope](#) of PUBLIC.

## 8.5.2.7 Data Encryption Status page

Table 3 specifies the format of the Data Encryption Status page.

**Table 3 — Data Encryption Status page**

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB) PAGE CODE (0020h) (LSB)							
1								
2	(MSB) PAGE LENGTH (n-3) (LSB)							
3								
4	I_T NEXUS SCOPE			Reserved		KEY SCOPE DATA ENCRYPTION SCOPE		
5	ENCRYPTION MODE							
6	DECRYPTION MODE							
7	ALGORITHM INDEX							
8	(MSB) KEY INSTANCE COUNTER (LSB)							
11								
12	Reserved	PARAMETERS CONTROL			VCELB	CEEMS		RDMD
13	Reserved							
23								
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 4.2.21.7 on page 57).

The KEY SCOPE DATA ENCRYPTION SCOPE field shall contain the value from the key scope data encryption scope in the saved data encryption parameters currently associated with the I\_T nexus on which this command was received (see 4.2.21.8 on page 58).

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 4.2.21.8 on page 58).

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 4.2.21.8 on page 58).

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 4.2.21.8 on page 58). 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 4.2.21.10 on page 59) assigned to the key indicated by the KEY SCOPE data encryption scope field value.

The PARAMETERS CONTROL field specifies information on how the data encryption parameters are controlled. The PARAMETERS CONTROL field values are specified in *table 4*.

**Table 4 — PARAMETERS CONTROL field values**

Code	Description
000b	Data encryption parameters control is not reported.
001b	Data encryption parameters are not exclusively controlled by external data encryption control.
010b	Data encryption parameters are exclusively controlled by the sequential-access device server.
011b	Data encryption parameters are not exclusively controlled by the automation/drive interface device server.
100b	Data encryption parameters are not exclusively controlled by a management interface.
101b-111b	Reserved

If the VCELB\_C bit is set to one in the Data Encryption Capabilities page, then the volume contains encrypted logical blocks (VCELB) bit shall be set to one when a mounted volume contains an encrypted logical block. The VCELB bit shall be set to zero if:

- a) the mounted volume does not contain any encrypted logical blocks;
- b) there is no volume mounted; or
- c) the VCELB\_C bit in the Data Encryption Capabilities page is set to zero.

The raw decryption mode disabled (RDMD) bit shall be set to one if the device server is configured to mark each encrypted record as disabled for raw read operations based on the RDMC\_C value and the raw decryption mode disable parameter in the saved data encryption parameters currently associated with the I\_T nexus on which the command was received (see 4.2.21.7 on page 57).

The check external encryption mode status (CEEMS) field shall contain the value from the check external encryption mode parameter in the saved data encryption parameters currently associated with the I\_T nexus on which the command was received (see 4.2.21.7 on page 57).

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 is 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 data encryption scope fields at the time the key was established in the device server. If more than one key associated descriptor is included, they shall be in 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 when the key was established in the device server. The AUTHENTICATED field is reserved. 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 when the key was established in the device server. The AUTHENTICATED field is reserved. 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 when the key was established in the device server. The AUTHENTICATED field is reserved. 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 when the key was established 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.

A metadata key-associated data descriptor (see 8.5.4.6 on 39) shall be included if the metadata key-associated data descriptor was included when the data encryption parameters were established. The KEY DESCRIPTOR field shall contain the M-KAD value associated with the key.

---



---

Editors Note 19 - KDB: No Changes in the rest of the clauses between here.

---



---

### 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 sent individually. An application client requests to send a page by using a SECURITY PROTOCOL OUT command with the SECURITY PROTOCOL field set to Tape Data Encryption security protocol and the SECURITY PROTOCOL SPECIFIC field set to the page code requested.

The SECURITY PROTOCOL SPECIFIC field (see table 5) specifies the type of page that the application client is sending.

**Table 5 — SECURITY PROTOCOL SPECIFIC field values**

Code	Description	Reference
0000h-000Fh	Reserved	
0010h	Set Data Encryption page	8.5.3.2 on 14
0011h	SA Encapsulation page	8.5.3.3 on 37
0012h-002Fh	Reserved	
0030h-003Fh	Restricted	ADC-3
0040h-FEFFh	Reserved	
FF00h-FFFFh	Vendor specific	

If the SECURITY PROTOCOL 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

## 8.5.3.2.1 Set Data Encryption page overview

Table 6 specifies the format of the Set Data Encryption page.

**Table 6 — Set Data Encryption page**

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

The PAGE LENGTH field specifies 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 7) specifies the [scope data encryption scope](#) of the data encryption parameters. Support for [data encryption](#) scope values of PUBLIC and ALL I\_T NEXUS are mandatory for device servers that support the Set Data Encryption page.

**Table 7 — 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	ALL I_T NEXUS	The data encryption parameters shall be shared with all I_T nexuses.
3-7		Reserved

See 4.2.21.11 on 60 for a description of the LOCK bit.

Table 8 describes the values for the check external encryption mode (CEEM) field..

**Table 8 — CEEM field values**

Code	Description
00b	Vendor specific
01b	Do not check the encryption mode that was in use when the block was written to the medium.
10b	On read and verify commands, check the encryption mode that was in use when the block was written to the medium. Report an error if the block was written in EXTERNAL mode (see 4.2.21.5 on page 54).
11b	On read and verify commands, check the encryption mode that was in use when the block was written to the medium. Report an error if the block was written in ENCRYPT mode (see 4.2.21.5 on page 54).

The device server shall terminate the SECURITY PROTOCOL OUT 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 CEEM field is set to either 10b or 11b, and:

- a) the DECRYPTION MODE field is set to DISABLE; or
- b) the EAREM bit in the algorithm descriptor (see *Editors Note 16 - KDB:*) for the algorithm specified by the ALGORITHM INDEX field is set to zero.

The raw decryption mode control (RDMC) field specifies if the device server shall mark each encrypted block written to the medium as disabled for read operations in raw mode (i.e., read operations with the decryption

mode set to RAW). The RDMC field shall be ignored if the ENCRYPTION MODE field is not set to ENCRYPT. *Table 9* specifies the values for the RDMC field when the ENCRYPTION MODE field is set to ENCRYPT.

**Table 9 — RDMC field values**

Code	Description
00b	The device server shall mark each encrypted block per the default setting for the algorithm ( <i>see table 129</i> ).
01b	Reserved
10b	The device server shall mark each encrypted block written to the medium in a format specific manner as enabled for raw decryption mode operations.
11b	The device server shall mark each encrypted block written to the medium in a format specific manner as disabled for raw decryption mode operations.

The device server shall terminate the command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST and the additional sense key set to INVALID FIELD IN PARAMETER DATA if:

- a) the ENCRYPTION MODE field is set to ENCRYPT;
- b) the RDMC field is set to 10b or 11b; and
- c) the RDMC\_C field in the algorithm descriptor for the encryption algorithm selected by the value in the ALGORITHM INDEX field is set to 1h, 6h, or 7h.

If the supplemental decryption key (SDK) bit is set to one, the key sent in this page shall be added to the set of data encryption parameters used by the device server for the selected scope. The KEY INSTANCE COUNTER shall not be incremented for supplemental decryption keys. The ENCRYPTION MODE and LOCK fields shall be ignored and the DECRYPTION MODE shall match the current setting for this scope. If the DECRYPTION MODE does not match the current settings for this scope, the device server shall terminate the command with CHECK CONDITION status and set the sense key to ILLEGAL REQUEST and the additional sense code to INVALID FIELD IN PARAMETER LIST.

If the physical device does not currently have a saved set of data encryption parameters associated with the I\_T nexus that sent the Set Data Encryption page or the [scope\\_data\\_encryption\\_scope](#) or decryption mode values do not match the values in that set of saved data encryption parameters, the device server shall terminate the command with CHECK CONDITION status and set the sense key to ILLEGAL REQUEST and the additional sense code to INVALID FIELD IN PARAMETER LIST.

If the SDK bit is set to one and the SDK\_C field is set to zero in the Data Encryption Algorithm descriptor field that matches the ALGORITHM INDEX in the Data Encryption capabilities page, the device server shall terminate the command with CHECK CONDITION status and set the sense key set ILLEGAL REQUEST and the additional sense code to INVALID FIELD IN PARAMETER LIST.

If the device server is processing a Set Data Encryption page with the SDK bit set to one and does not have the resource available to store this key the device server shall terminate the command with CHECK CONDITION status and set the sense key to ILLEGAL REQUEST and the additional sense code to MAXIMUM NUMBER OF SUPPLEMENTAL DECRYPTION KEYS EXCEEDED. Any previously saved supplemental decryption keys shall not be affected by this error.

If the SDK bit is set to zero, the key sent in this page shall be the key used for both encryption and decryption. Any keys that have been previously stored by the device server shall be removed from memory. See 4.2.21.6 on 56.

If the clear key on demount (CKOD) bit is set to one the physical device shall set the data encryption parameters to default values upon completion of a volume demount. If the CKOD bit is set to zero, the demounting of a volume shall not affect the data encryption parameters. If the CKOD bit is set to one and there is no volume mounted, the device server shall terminate the command with CHECK CONDITION status and set the sense key to ILLEGAL REQUEST and the additional sense code to INVALID FIELD IN PARAMETER DATA.



If the clear key on reservation preempt (CKORP) bit is set to one the physical device shall set the data encryption parameters to default values when a persistent reservation is preempted (i.e., a PERSISTENT RESERVE OUT command specifying a service action of PREEMPT or PREEMPT AND ABORT is processed). If the CKORP bit is set to zero, a preemption of a persistent reservation shall not affect the data encryption parameters. If the CKORP 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 and set the sense key to ILLEGAL REQUEST and the additional sense code to INVALID FIELD IN PARAMETER DATA.

If the clear key on reservation loss (CKORL) bit is set to one the physical device shall set the data encryption parameters to default values on a reservation loss (see 3.1.56 on page 7). 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 and set the sense key to ILLEGAL REQUEST and the additional sense code to INVALID FIELD IN PARAMETER DATA.

Table 10 specifies the values for the ENCRYPTION MODE field.

**Table 10 — ENCRYPTION MODE field values**

Code	Name	Description
0h	DISABLE	Data encryption is disabled.
1h	EXTERNAL	The data associated with the 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) command using the algorithm specified in the ALGORITHM INDEX field and the key specified in the KEY field.
3h-Fh		Reserved

Table 11 specifies the values for the DECRYPTION MODE field. See 4.2.21.3 on 53 for configuration and exception condition requirements.

**Table 11 — DECRYPTION MODE field values**

Code	Name	Description
0h	DISABLE	Data decryption is disabled. If the device server encounters an encrypted logical block while reading, it shall not allow access to the data.
1h	RAW	Data decryption is disabled. If the device server encounters an encrypted logical 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 when processing 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 specified in the KEY field.
3h	MIXED	The device server shall decrypt all data that is read from the medium that the device server determines was encrypted when processing 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 specified 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 physical device 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. If the algorithm specified in the ALGORITHM INDEX field is disabled, 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 ENCRYPTION ALGORITHM DISABLED.

If a volume is mounted and the combination of the ENCRYPTION MODE, DECRYPTION MODE, and ALGORITHM INDEX fields is not valid for the mounted volume or current logical position, 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 KEY FORMAT field specifies the format of the value in the KEY field. Values for the KEY FORMAT field are specified in *table 12*.

**Table 12 — KEY FORMAT field values**

Code	Description	Reference
00h	The KEY field contains the key to be used to encrypt or decrypt data.	<i>b) on 20</i>
01h	The KEY field contains a vendor-specific key reference.	<i>8.5.3.2.3 on 33</i>
02h	The KEY field contains the key wrapped by the device server public key.	<i>8.5.3.2.4 on 34</i>
03h	The KEY field contains a key that is encrypted using ESP-SCSI.	<i>8.5.3.2.5 on 36</i>
04h-BFh	Reserved	
C0h-FFh	Vendor specific	

---



---

Editors Note 20 - DAP: Would like to use the term logical block instead of data throughout the security text. Make the change but keep in mind it may be better to use encrypted block in some instances.

---



---

The KEY LENGTH field specifies the length of the KEY field in bytes.

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 the ENCRYPTION MODE field is set to EXTERNAL 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 written using the data encryption parameters established by this command.

If more than one key-associated data descriptor is specified in the Set Data Encryption page, they shall be in increasing numeric order of the value in the DESCRIPTOR TYPE 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 if:

- a) key-associated descriptors are included in the KEY-ASSOCIATED DATA DESCRIPTORS LIST field;
- b) DECRYPTION MODE field is not set to RAW; and;
- c) the ENCRYPTION MODE field is not set to:
  - A) EXTERNAL; or
  - B) ENCRYPT.

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 is reserved. The KEY DESCRIPTOR field shall contain the U-KAD value associated with the encrypted block. 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 UKADF bit is set to one in the data encryption algorithm descriptor, the ENCRYPTION MODE field is set to ENCRYPT, and:

- a) the length of the KEY DESCRIPTOR field is not equal to the value in the MAXIMUM UNAUTHENTICATED KEY-ASSOCIATED BYTES field of the data encryption algorithm descriptor; or
- b) the parameter data does not contain an unauthenticated key-associated data descriptor.

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 is reserved. The KEY DESCRIPTOR field shall contain the A-KAD value associated with the encrypted

block. 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 AKADF bit is set to one in the data encryption algorithm descriptor, the ENCRYPTION MODE field is set to ENCRYPT, and:

- a) the length of the KEY DESCRIPTOR field is not equal to the value in the MAXIMUM AUTHENTICATED KEY-ASSOCIATED BYTES field of the data encryption algorithm descriptor; or
- b) the parameter data does not contain an authenticated key-associated data descriptor.

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-ASSOCIATED DATA SET. If a nonce value descriptor is included, the AUTHENTICATED field is reserved. The KEY DESCRIPTOR field shall contain the nonce value associated with the encrypted block.

A metadata key-associated data descriptor (see 8.5.4.6 on 39) may be included if the DECRYPTION MODE field is set to RAW and the encryption algorithm requires any metadata key-associated data to be associated with encrypted logical blocks read when the DECRYPTION MODE field is set to RAW.

A metadata key-associated data descriptor (see 8.5.4.6 on 39) shall be included if the ENCRYPTION MODE field is set to EXTERNAL and the encryption algorithm requires any metadata key-associated data to be associated with logical blocks written when the ENCRYPTION MODE field is set to EXTERNAL.

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 an M-KAD is included and:

- a) the ENCRYPTION MODE field is not set to EXTERNAL and the DECRYPTION MODE field is not set to RAW; or,
- b) the encryption algorithm specified by the ALGORITHM INDEX field does not support M-KAD.