To: INCITS T10 Committee

From: Paul Entzel, Quantum

Date: 16-November 2005

Document: T10/05-446r0

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

1 Revision History

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

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 application 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 two new commands to the SSC-3 command set.

The DATA SECURITY IN command is a Service Action derivative of the SERVICE ACTION IN (12) command. This command can request several reports containing supported features and enabled modes.

The DATA SECURITY OUT command is a derivative of the SERVICE ACTION OUT (12) command. This command 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.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 the device server.

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

3.2 Addition to the model clause (4)

3.2.1 Additions to subclause 4.2.17 Reservations

Add to table 12 the following commands:

Command	**Addressed LU has this type of persistent reservation held by another I_T nexus**				
	From any I_T nexus		**From registered I_T nexus (RR all types)**	**From I_T nexus not registered**	
	Write Exclusive	**Exclusive Access**		**Write Exclusive - RR**	**Exclusive Access - RR**
DATA SECURITY IN	Conflict	Conflict	Allowed	Allowed	Conflict
DATA SECURITY OUT	Conflict	Conflict	Allowed	Conflict	Conflict

3.2.2 Additional subclause

Add the following subclause to clause 4:

4.2.19 Data Encryption

4.2.19.1 Data Encryption overview

An SSC-3 compliant device may contain hardware or software that is capable of encrypting the data within logical blocks to provide security against unauthorized access to that data. The DATA SECURITY IN and DATA SECURITY OUT commands provide a means for the application client to monitor and control the encryption process within the device. A device server that supports the DATA SECURITY OUT command shall also support the DATA SECURITY IN command.

The DATA SECURITY OUT command is used to enable and disable encryption mode and set the encryption key. The DATA SECURITY IN command is used to discover the type of data security features supported by the device server and the current configuration of data security features.

4.2.19.2 Encrypting data on the medium

The application client controls the data encryption process by use of the DATA SECURITY OUT command. Data encryption is enabled after the device server successfully processes a DATA SECURITY OUT command that sends a Set Encryption Key page with the encrypt bit set to one and a valid key. Data encryption is disabled for an I_T nexus after:

a) a DATA SECURITY OUT command from the I_T nexus with a Set Encryption Key page is processed successfully with the ENCRYPT bit set to zero;

b) the volume is dismounted;

c) the CKORL bit was set to one in the Set Encryption Key page that enabled encryption and the I_T nexus that had enabled encryption losses its reservation;

d) the key with a scope of GLOBAL or RESERVATION GROUP being used by the I_T nexus is cleared; or

e) a hard reset

While the data encryption process is enabled, all data received by the device server as part of a WRITE(6) or WRITE(16) command shall be encrypted before being recorded on the medium. Filemarks written due to a WRITE FILEMARKS(6) or WRITE FILEMARKS(16) command shall not be affected by the encryption process.

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. Encrypted blocks shall have no impact on space or locate commands.

A device server that supports encryption should be able to distinguish encrypted blocks from unencrypted blocks. If the device server is able to distinguish encrypted blocks from unencrypted blocks, an attempt to read or verify an encrypted block when decryption has not been enabled shall cause the device server to terminate the command with CHECK CONDITION status, with the Sense Key set to DATA PROTECT, and the addition sense code set to UNABLE TO DECRYPT DATA. The device server shall establish the logical position at the BOP side of the encrypted block.

Editor’s note: UNABLE TO DECRYPT DATA is a new ASC.

If the device server is able to distinguish encrypted blocks from unencrypted blocks, an attempt to read or verify an unencrypted block when the DECRYPT bit is set to one and the RUD bit is set to one shall cause the device server to terminate the command with CHECK CONDITION status, with the Sense Key set to DATA PROTECT, and the addition sense code set to UNENCRYPTED DATA ENCOUNTER WHILE DECRYPTING. The device server shall establish the logical position at the BOP side of the unencrypted block.

Editor’s note: UNENCRYPTED DATA ENCOUNTER WHILE DECRYPTING is a new ASC.

A device server that supports encryption and has been configured to decrypt the data may be able to determine if the encryption key is correct for an encrypted block. If the device server is able to determine if the encryption key is correct, an attempt to read or verify an encrypted block when decryption is enabled but the encryption key is incorrect for the block shall cause the device server to terminate the command with CHECK CONDITION status, with the Sense Key set to DATA PROTECT, and the addition sense code set to INCORRECT DATA ENCRYPTION KEY. The device server shall establish the logical position at the BOP side the encrypted block.

Editor’s note: INCORRECT DATA ENCRYPTION KEY is a new ASC.

4.2.19.4 Exhaustive-search attack prevention

To prevent an exhaustive-search attack from discovering the encryption key, the device server shall provide a mechanism to prevent unlimited attempts at setting a data encryption key and then attempting to read the data.

If the device server is not capable of distinguishing encrypted data from unencrypted data or is not capable of authenticating the encryption key, it should limit the key changes by introducing a delay in the processing of the DATA SECURITY OUT command that will slow the key change process. The delay should be sufficient so that trying all possible encryption keys will take so many years that it becomes impractical.

If the device server is capable of distinguishing encrypted data from unencrypted data and is capable of authenticating the data encryption key, it shall use one of the following techniques:

a) introducing a delay in the processing of the DATA SECURITY OUT command that will slow the key change process;

b) limiting the number if attempts at changing the key and reading encrypted data per mount of the volume; or

c) some other vendor unique process that limits the ability to discover the encryption key using and exhaustive-search approach.

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 subsequence DATA SECURITY OUT commands with the PAGE CODE field set to the data encryption page and the DECRYPT bit set to one shall be terminated with CHECK CONDITION status, 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 medium is dismounted from the device or a hard reset event.

Editor's note: DATA DECRYPTION KEY FAIL LIMIT REACHED is a new ASC.

4.2.19.5 Managing keys within the device server

The security provided by data encryption is only as good and the security used when managing the keys. For this reasons, the data encryption key and mode are volatile in the device server and never reported to an initiator. The device server also may have limited resources for storage of keys.

If a device server processes a Set Data Encryption page with the ENCRYPT and DECRYPT bits both set to zero, the device server shall release any resources that it had allocated to store a key value for the I_T nexus associated with the DATA SECURITY OUT command and shall clear any memory containing the key value. A unit attention condition with the additional sense of DATA ENCRYPTION MODE CHANGED BY ANOTHER INITIATOR shall be establish for any other I_T nexus that is affected by the loss of the key, that is, any I_T nexus that is using a scope of PUBLIC and sharing the key.

Editor's note: DATA ENCRYPTION MODE CHANGED BY ANOTHER INITIATOR is a new ASC.

If a device server processes a Set Data Encryption page with the ENCRYPT bit or the DECRYPT bit set to one, the device server shall release any resources that it had allocated to store a key value set by a previous DATA SECURITY OUT command from that I_T nexus and shall clear any memory containing the key value. A unit attention condition with the additional sense of DATA ENCRYPTION MODE CHANGED BY ANOTHER INITIATOR shall be establish 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 scope of PUBLIC and sharing the key).

A device server shall save at most one key with a scope of GLOBAL. If a device server processes a Set Data Encryption page with the SCOPE field set to GLOBAL, the device server shall release any resources that it had allocated to store a key value set by a previous DATA SECURITY OUT command with a scope value of GLOBAL and shall clear any memory containing the key value. A unit attention condition with the additional sense of DATA ENCRYPTION MODE CHANGED BY ANOTHER INITIATOR shall be establish for any other I_T nexus that is affected by the change of the key, that is, any I_T nexus that is using a scope of public and sharing the key.

A device server shall key at most one key with a 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 release any resources that it had allocated to store a key value set by a previous DATA SECURITY OUT command with a scope value of RESERVATION GROUP and shall clear any memory containing the key value. A unit attention condition with the additional sense of DATA ENCRYPTION MODE CHANGED BY ANOTHER INITIATOR shall be establish for any other I_T nexus that is affected by the change of the key, that is, any I_T nexus that is using a scope of public and sharing the key.

4.2.19.6 Encryption mode verification within the device server

There are conditions outside of the control of an application client which cause the device server to stop encrypting once configured to encrypt (see 4.2.19.2). If the application client sets the ENCRYPTE bit to one in the WRITE(6) and WRITE(16) commands, the device server shall confirm that it is still configured to encrypt the data. If it is not, the command is rejected. This provides the application client an opportunity to re-configure the device server to encrypt without writing any unencrypted data to the medium.

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
DATA SECURITY IN	O	A3h/XXh	No	7.X
DATA SECURITY OUT	O	A4h/XXh	No	7.Y

3.3.2 Addition to subclause 5.6 WRITE(16)

In subclause 5.6 (WRITE(16)), add a bit field to table 18 called ENCRYPTE in byte 1, bit 1.

Add a paragraph to describe this bit:

If the encryption expected (ENCRYPTE) bit is set to one and the device server does not support encryption, the command shall be terminated with CHECK CONDITION status, the Sense Key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB. If the ENCRYPTE bit is set to one and the device server is not currently configured to encrypt the data before writing it to the medium, the command shall be terminated with CHECK CONDITION status, the Sense Key set to ILLEGAL REQUEST, and the additional sense code set to ENCRYPTION CONFIGURATION MISMATCH. If the ENCRYPTE bit is set to one and the device server is currently configured to encrypt the data before writing it to the medium, or the ENCRYPT bit is set to zero, the command shall be processed normally.

Editor's note: ENCRYPTION CONFIGURATION MISMATCH is a new ASC.

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
DATA SECURITY IN	O	A3h/XXh	No	7.X
DATA SECURITY OUT	O	A4h/XXh	No	7.Y

3.4.2 Addition to subclause 6.8 WRITE(6)

In subclause 6.8 (WRITE(6)), add a bit field to table 28 called ENCRYPTE in byte 1, bit 1.

Add a paragraph to describe this bit:

If the encryption expected (ENCRYPTE) bit is set to one and the device server does not support encryption, the command shall be terminated with CHECK CONDITION status, the Sense Key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB. If the ENCRYPTE bit is set to one and the device server is not currently configured to encrypt the data before writing it to the medium, the command shall be terminated with CHECK CONDITION status, the Sense Key set to ILLEGAL REQUEST, and the additional sense code set to ENCRYPTION CONFIGURATION MISMATCH. If the ENCRYPTE bit is set to one and the device server is currently configured to encrypt the data before writing it to the medium, or the ENCRYPT bit is set to zero, the command shall be processed normally.

3.5 Additions to common commands clause (7)

7.X. DATA SECURITY IN command

7.X.1 DATA SECURITY IN command description

The DATA SECURITY IN (see table A1) 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.

Table A1 –DATA SECURITY IN command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (A3h)							
1	Reserved			SERVICE ACTION				
2	PAGE CODE							
3	(MSB)	Reserved						
5								(LSB)
6	(MSB)	ALLOCATION LENGTH						
9								(LSB)
10	Reserved							
11	CONTROL							

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

Table B1 – PAGE CODE field values

Value	Description	Reference
00h	Supported DATA SECURITY IN pages	7.X.2
01h	Supported DATA SECURITY OUT pages	7.X.3
02h – 0Fh	Reserved	
10h	Data encryption capabilities page	7.X.4
11h	Data encryption status page	7.X.5
12h - FFh	Reserved	

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

7.X.2 Supported DATA SECURITY IN pages

Table C1 shows the format of the Supported DATA SECURITY IN pages report.

Table C1 – Supported DATA SECURITY IN pages report

Bit Byte	7	6	5	4	3	2	1	0
0	PAGE CODE (00h)							
1	Reserved							
2	(MSB)	PAGE LENGTH (n-3)						
3								(LSB)
4	(MSB)	SUPPORTED PAGE LIST						
n								(LSB)

The SUPPORTED PAGE LIST field shall contain a list of all of the pages that the device server supports for the DATA SECURITY IN command in numerical order starting with 00h.

7.X.3 Supported DATA SECURITY OUT pages

Table D1 shows the format of the Supported DATA SECURITY OUT pages report.

Table D1 – Supported DATA SECURITY OUT pages report

Bit Byte	7	6	5	4	3	2	1	0
0	PAGE CODE (01h)							
1	Reserved							
2	(MSB)	PAGE LENGTH (n-3)						
3								(LSB)
4	(MSB)	SUPPORTED PAGE LIST						
n								(LSB)

The SUPPORTED PAGE LIST field shall contain a list of all of the pages that the device server supports for the DATA SECURITY OUT command in numerical order.

7.X.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	PAGE CODE (10h)							
1	Reserved							
2	(MSB)	PAGE LENGTH (n-3)						
3								(LSB)
	Encryption algorithm descriptors							
4	(MSB)	Data encryption algorithm descriptor						
								(LSB)
	:							
	(MSB)	Data encryption algorithm descriptor						
s								(LSB)

See SPC-3 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 (n-3)						
3								(LSB)
4	Reserved		AK_C	DED_C	DECRYPT_C		ENCRYPT_C	
5	Reserved				IV_RN	IV_EOU	IV_WPU	IV_MU
6	Reserved							
9								
10	ALGORITHM NAME LENGTH (n-11)							
11								
12	ALGORITHM NAME							
n								

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 DATA SECURITY 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

Value	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

Value	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 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 authenticate key capable (AK_C) bit shall be set to one if the device server is capable of authenticating the data encryption key provided to decrypt the data. The AK_C bit shall be set to zero if the device server is not capable of authenticating the data encryption key provided to decrypt the data. If the ability to authenticate the key is format specific and a volume is mounted, the AK_C shall be set based on the current format of the medium. If no volume is mounted, the AK_C bit shall be set to one if the device server is capable of authenticating the key in any format that the device server supports.

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

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

The ALGORITHM NAME LENGTH contains the length in bytes of the ALGORITHM NAME field.

The ALGORITHM NAME field contains a null terminated, null padded UTF-8 format string that describes the encryption algorithm. The string shall contain the standardization authority that has registered the algorithm if there is a standard that defines it.

7.X.5 Data encryption status page

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

Table S1 – Data encryption status page

Bit Byte	7	6	5	4	3	2	1	0
0	PAGE CODE (11h)							
1	Reserved							
2	(MSB)	PAGE LENGTH (n-3)						
3								(LSB)
4	Reserved					RUD	DECRYPT	ENCRYPT
5	SCOPE			Reserved		SOURCE		
6	(MSB)	Reserved						
n								(LSB)

The ENCRYPT bit shall be set to one if the device server is enabled to encrypt data received from the I_T nexus as part of a WRITE(6) or WRITE(16) command before writing it to the medium. The ENCRYPT bit shall be set to zero if the device server is not enabled to encrypt data received from the I_T nexus as part of a WRITE(6) or WRITE(16) command before writing it to the medium.

The DECRYPT bit shall be set to one if the device server is enabled to decrypt data read from the medium while processing a read or verify command from this I_T nexus. The decrypt bit shall be set to zero if the device server is not enabled to decrypt data read from the medium while processing a read or verify command from this I_T nexus.

The report unencrypted data (RUD) bit shall be set to one if the device server is configured to report an error if it encounters unencrypted data while processing a read or verify command from this I_T nexus. The RUD bit shall be set to zero if the device server is not configured to report an error if it encounters unencrypted data while processing a read or verify command from this I_T nexus.

The scope field (see table E2) indicates the scope of the data encryption key and mode set by this I_T nexus.

Table E2 - SCOPE field values

Value	Description
0	The data encryption key and mode are default values.
1	The data encryption key and mode are unique to this I_T nexus.
2	The data encryption key and mode set by this I_T nexus are shared with all I_T Nexus that share in a reservation for the logical unit.
3	The data encryption key and mode set by this I_T nexus are shared with all other I_T nexuses.
4 – 7	Reserved

The SOURCE field (see table E3) indicates the source of the encryption key and data encryption mode for this I_T nexus.

Table E3 - SOURCE field values

Value	Description
0	The data encryption key and mode are default values.
1	The data encryption key and mode are unique to this I_T nexus.
2	The data encryption key and mode were established by another I_T nexus and are being shared with other reservation holds.
3	The data encryption key and mode were established by another I_T nexus and are being shared globally.
4- 7	Reserved

7.Y. DATA SECURITY OUT command

7.Y.1 DATA SECURITY OUT command description

The DATA SECURITY OUT (see table A2) is used to configure the data security methods in the device server and on the medium. The command supports a series of pages, only one of which may be sent with each command.

Table A2 –DATA SECURITY OUT command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (A4h)							
1	Reserved			SERVICE ACTION				
2	PAGE CODE							
3	(MSB)	Reserved						
5								(LSB)
6	(MSB)	PARAMETER LIST LENGTH						
9								(LSB)
10	Reserved							
11	CONTROL							

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

Table B2 – PAGE CODE field values

Value	Description	Reference
00h – 0Fh	Reserved	
10h	Set data encryption page	7.Y.2
11h - FFh	Reserved	

The parameter list length field specifies the length in bytes of parameter data to be transferred from the application client to the device server. If the parameter length does not match the length of the specified page, 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 PARAMETER LIST LENGTH ERROR.

7.Y.2 Set data encryption page

Table Y1 shows the parameter list format of the set data encryption page.

Table Y1 – Set Data Encryption page

Bit Byte	7	6	5	4	3	2	1	0
0	PAGE CODE (10h)							
1	Reserved							
2	(MSB)	PAGE LENGTH (n-3)						
3								(LSB)
4	SCOPE			Reserved				CKORL
5	Reserved					RUD	DECRYPT	ENCRYPT
6	ALGORITHM INDEX							
7	KEY FORMAT							
8	(MSB)	Reserved						
17								(LSB)
18	(MSB)	KEY LENGTH (n-19)						
19								(LSB)
20		KEY						
n								

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 mode and key.

Table Y2 – SCOPE field values

Value	Name	Description
0	PUBLIC	The data encryption mode and key shall be ignored. The I_T nexus shall use values that are shared by other I_T nexuses. If no I_T nexuses are sharing values, the device server shall use default values.
1	LOCAL	The data encryption mode and key are unique to the I_T nexus associated with the DATA SECURITY OUT command and shall not be shared with other I_T nexuses.
2	RESERVATION GROUP	The data encryption mode and key shall be shared with all participants in a reservation.
3	GLOBAL	The data encryption mode and key shall be shared with all I_T nexuses.
4 – 7		Reserved

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

1. If the scope for the I_T nexus is not PUBLIC, the values set by a DATA SECURITY OUT command associated with the I_T nexus; or
2. If the scope for the I_T nexus is PUBLIC:
   1) If the I_T nexus is participating in a reservation for the logical unit, the values set by another participant in the reservation with a scope of RESERVATION GROUP;
   2) the values set by anther I_T nexus with a scope of GLOBAL; or
   3) the default values.

If the clear key on reservation loss (CKORL) bit is set the device server shall set the encryption key and encryption mode to default values on the loss of a reservation. If the CKORL bit is set to zero, the loss of a reservation shall not affect the encryption key or encryption mode. If the CKORL bit is set to one and there is no reservation in affect for the I_T nexus associated with the DATA SECURITY 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.

The bit fields in byte 5 and the value in the ALGORITHM INDEX field are collectively referred to as the encryption mode.

If the ENCRYPT bit is set to one, 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.

If the ENCRYPT bit is set to one 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 DECRYPT bit is set to one, the device server shall decrypt all data that is read from the medium before sending it in response to a READ(6) or READ(16) 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 DECRYPT bit is set to one 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 report unencrypted data (RUD) bit is set to one and the device server encounters unencrypted data when processing a READ(6), READ(16), VERIFY(6), or VERIFY(16) command, 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 UNENCRYPTED DATA ENCOUNTERED WHILE DECRYPTING. The device server shall leave the medium positioned in front of the unencrypted block.

Editor's note: UNENCRYPTED DATA ENCOUNTERED WHILE DECRYPTING is a new ASC.

If the RUD bit is set to one and the DECRYPT bit 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 DATA SECURITY 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 Y3.

Table Y3 – KEY FORMAT field values

Value	Description
00h	Key is in plain text
01h – FFh	Reserved

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

The KEY field contains the encryption key to use when encrypting and decrypting data.