To: INCITS Technical Committee T10
From: Kevin Butt
Date: February 14, 2007 10:29 am
Document: T10/07-005r0 — SSC-3: Encryption unsupported medium

1. Revisions

1. 07-005r0 Initial version
2. 07-005r1 Incorporated feedback from Paul Entzel and added a removal of “a volume is mounted that does not support data encryption using the algorithm specified by the algorithm index in the data encryption parameter“ that was inadvertantly left out of revision 0. Incorporated modifications from 14 Feb telecon.

2. Introduction

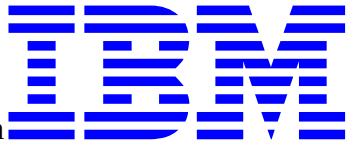
There is no description in SSC-3 of what the behavior should be if a volume that does not support encryption is inserted into a drive that has been configured to encrypt data with a valid set of encryption parameters and the CKOD bit set to zero.

There is a status bit to indicate the algorithm validity for the mounted volume in Table 99 — Data Encryption Algorithm:

> The algorithm valid for mounted volume (AVFMV) bit shall be set to one if there is a volume currently mounted in the device and the encryption algorithm being described is valid for that volume. The AVFMV bit shall be set to zero if there is no volume mounted in the device or the algorithm is not valid for the currently mounted volume.

The text related to clearing the data encryption parameters is in SSC-3r03b clause 4.2.20.5 Managing keys within the device server:

> 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 demounted;
>
> 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 ALL I_T NEXUS in the saved data encryption parameters and the device server experiences a reservation loss (see 3.1.55);
>
> 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 volume is mounted that does not support data encryption using the algorithm specified by the algorithm index in the data encryption parameter;



f) a microcode update is performed on the device;

g) a power on condition occurs; or

h) other vendor-specific events.

and later in the same section:

> 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 4.2.20.6) and is affected by the change of the key.

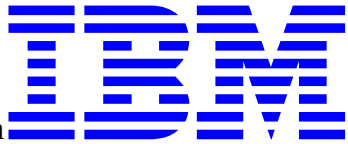
The solution to this problem is on an attempted write to a volume that does not support the selected algorithm at the current location ~~(or at all)~~ return a Check Condition to the host with a new additional sense code indicating that the volume mounted does not support the current algorithm at the current position. For any write when in this condition continue reporting this Check Condition until the encryption parameters are changed to a supported algorithm, encryption is disabled, or the logical position is changed to a position that does support the selected algorithm. (*This gives positive indication to the host that there is a problem that needs to be resolved. This allows the volume to be mounted before encryption parameters are modified and to not have an automatic Check Condition on load or other unaffected operations. There is only reason to return a Check Condition on the attempt to write. All other rules for read, etc. apply. If the host attempting to do the writing is not encryption aware — that is, is not the one controlling the encryption settings — it could get locked out of the drive. This is still better than not encrypting data that was supposed to be encrypted. This is the same issue for many error responses in this type of environment.*)

Additionally, there should be no restrictions on setting a key based on algorithm (if it is known to a drive it should be settable, even if it does not apply to this volume). This allows more flexible third-party setup of encryption. The above method will catch/notify and prevent unintended usage/behavior.

Additionally, in some formats, a volume must be formatted for encryption (e.g. first write from BOP determines the format of the medium) or non-encryption. There needs to be a means to determine if the volume supports encryption at the current logical position or not (specifically at BOP vs. away from BOP).

For example, a previous generation drive does not support encryption. A current generation drives supports encryption and also supports reading and writing to the previous generation drive. However, in order to encrypt the format needs to be changed and this can only be done when positioned at BOP (and subsequently all existing data is overwritten). A volume that was written on the previous generation drive is loaded into the encrypting drive. If the volume is positioned at BOP and it is not write protected, it is legal for the encryption capable drive to write in either the legacy non-encrypting format or in the encryption capable format. However, if the volume is positioned away from BOP it can only be written to in the already written legacy format.

To: INCITS Technical Committee T10
From: Kevin Butt
Date: February 14, 2007 10:29 am
Document: T10/07-005r0 — SSC-3: Encryption unsupported medium



Applications need to be able to determine when they can use encryption and when they cannot. We propose adding another status bit to the Data encryption algorithm descriptor that provides this feedback.

3. Proposal

4.2.20.2 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 4.2.20.6), 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 (see 4.2.20.7).

If data encryption is enabled for an I_T nexus and the mounted volume supports the selected encryption algorithm at the current logical position, all ~~data~~ logical blocks 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~~ are logical objects that shall not be encrypted.

If data encryption is enabled for an I_T nexus and the mounted volume does not support the selected encryption algorithm at the current logical position, the receipt of a WRITE(6) or WRITE(16) command 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 ENCRYPTION PARAMETERS NOT USEABLE.

NOTE: ENCRYPTION PARAMETERS NOT USEABLE is a new additional sense code. Should be 74xx.

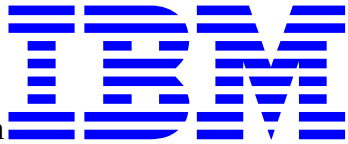
If data encryption is enabled for an I_T nexus and the mounted volume does not support the selected encryption algorithm at the current logical position, the receipt of a WRITE FILEMARKS (6) or WRITE FILEMARKS(16) command may or may not cause the device server to terminate the command with CHECK CONDITION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

4.2.20.3 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. The decryption mode shall be ignored when processing a filemark during a read or verify command.

.

.



.

4.2.20.4 Exhaustive-search attack prevention

.

.

.

4.2.20.5 Managing keys within the device server

.

.

.

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 ALL I_T NEXUS in the saved data encryption parameters and the device server experiences a reservation loss (see 3.1.55);

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 volume is mounted that does not support data encryption using the algorithm specified by the algorithm index in the data encryption parameter;~~

f) a microcode update is performed on the device;

g) a power on condition occurs; or

h) 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:

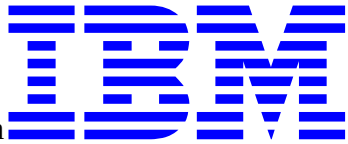
.

.

.

8.5.2.4 Data Encryption Capabilities page

.



.

.

In Table 99 — Data Encryption Algorithm descriptor, add an AVFCLP bit in bit 7 and 6 of byte 5.

Add the following description of the AVFLP bit.

The algorithm valid for current logical position (AVFCLP) field defines if the encryption algorithm being described is valid for writing to the mounted volume at the current logical position. Table x defines

TABLE 99+1. Definition of algorithm valid for current logical position field

Value	Description
00b	Current logical position has no bearing on algorithm validity.
01b	The encryption algorithm being described is not valid for writing to the mounted volume at the current logical position.
10b	The encryption algorithm being described is valid for writing to the mounted volume at the current logical position.
11b	Reserved