From: Gerry Houlder, Seagate Technology <gerry.houlder@seagate.com>
Subj: SBC-3 Model for encrypting disk drives
Date: June 26, 2008

Overview

There are a number of efforts under way to create protocols for doing authorization and enabling or disabling features of an encrypting disk drive. These efforts may not agree on the methods for doing these operations, but there is a need to standardize the resulting drive behavior when these features are enabled or disabled. This proposal is a start for that effort.

This text is all new and partly based on behavior defined for encrypting tape drives.

SBC-3 changes:

4.22 Data locking and encryption

4.22.1 Data locking and encryption overview
A device compliant with this standard may contain hardware or software that is capable of performing device locking and/or encrypting and decrypting the data within logical blocks (i.e., user data) to provide security against unauthorized access to that data. The SECURITY PROTOCOL IN and SECURITY PROTOCOL OUT commands (see SPC-4) specifying a security protocol provide a means for the application client to monitor and control the locking, 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 means to control the locking, encryption, and decryption processes are specific to the underlying security protocols of the SECURITY PROTOCOL IN and SECURITY PROTOCOL OUT commands. Although these processes are different for various security protocols, the resulting behavior should be compliant with the following clauses.

4.22.2 Device locking
Device locking is when an application client is not allowed to perform certain SCSI commands (e.g., read or write commands). An application client is required to perform a security protocol specific authorization process to unlock the device. If the device also supports encryption, this authorization process also authorizes use of the proper encryption key for encrypting and decrypting user data.

A device may support a single locking range that encompasses the entire LBA range of the logical unit or a device may support multiple locking ranges. A locking range is a range of LBAs with a starting and ending address. Locking ranges should not overlap each other, so each LBA is in only one locking range. If multiple locking ranges are supported, the application client may be authorized to access some locking ranges but not others. If encryption is supported, separate locking ranges should have separate encryption keys.

Some security protocols may allow separate read lock and write lock capability for each locking range.

If any I_T nexus unlocks a locking range, then it should be unlocked for all other I_T nexuses as well.

If the application client sends a command with a starting LBA that is in a locked LBA range, then no data is transferred and the command is terminated with CHECK CONDITION status with the sense key set to DATA PROTECT and the additional sense code set to ACCESS DENIED – NO ACCESS RIGHTS.

If the application client sends a command with a starting LBA that is in an unlocked LBA range but the transfer length extends into a locked LBA range, then no data is transferred and the command is terminated with CHECK CONDITION status with the sense key set to DATA PROTECT and the additional sense code set to ACCESS DENIED – NO ACCESS RIGHTS.

If the device is in an unlocked state, it should become locked if a power cycle occurs. Devices that support multiple locking ranges may allow some ranges to remain unlocked if a power cycle occurs. There shall be a security protocol specific process to lock or unlock the device at any time.

4.22.3 Encrypting data on the medium

For write operations an encrypting device receives plain text data from an application client, encrypts the data, and saves the encrypted data on the medium. For read operations an encrypting device reads the encrypted data from the medium, decrypts the data, and returns plain text data to the application client. If both locking and encryption are supported, read and write operations are only allowed in unlocked LBA ranges.

The encryption key for any LBA range shall only be changeable by a security protocol specific process.

If data is written (i.e., encrypted) with one key value but read back (i.e., decrypted) with a different key value, the plain text data that was written will not be the result. The data is transferred and the command is terminated with GOOD status and there is no indication that incorrect data has been transferred.

4.22.4 Cryptographic erase

If the SCSI device supports encryption, it shall also provide a cryptographic erase capability. Cryptographic erase is a process in which the encryption key(s) for the entire device or an LBA range on the device are eradicated and replaced with different randomly generated key(s). This has the effect of making the original plain text data undecipherable (within the limits of an application client being able to guess the original encryption key and encryption algorithm). This feature is invoked through a security protocol specific process.

The purpose of cryptographic erase is to securely make the original plain text user data unrecoverable. An advantage of cryptographic erase is speed. An entire device is made undecipherable in only a few seconds, when it might take hours to overwrite all of the data on a SCSI device using a process like the FORMAT or WRITE SAME command.

When the medium in a device is cryptographically erased, all logical blocks on the medium need to be rewritten before valid data can be read from those logical blocks.

4.22.5 Encryption and protection information interaction

A SCSI device may support both encryption and protection information features. In this configuration the protection information bytes are encrypted along with the user data.

If the encryption key is changed (e.g., the device is cryptographically erased) and the data is not rewritten with the new encryption key, then a read command with protection information checking enabled that is sent to the SCSI device should terminate with CHECK CONDITION status with

the sense key set to ABORTED COMMAND and the sense code set to one of the options described in table 39 [editors note: the RDPROTECT field table]. Repeating the read command with protection information checking disabled should end with GOOD status but the returned data is invalid because it was decrypted with the wrong decryption key. All logical blocks affected by the encryption key change need to be rewritten to restore normal operation.

[Note: SPC-4 editor please verify that the ASCs indicated here are marked as allowed for direct access and optical devices in the ASC tables.]