To: INCITS Technical Committee T10
From: Kevin Butt
Date: August 27, 2007 1:00 pm
Document: T10/07-374r0 — SSC-3: End-to-end Logical Block Protection

1. Revisions

07-374r0: Initial revision (August 16, 2007) using SSC-3r03d as base.

2. Introduction

KEY:

~~Deleted Text~~

Added Text

Updates to added text

EDITOR'S NOTE: <Text>

Questions

Please see document 07-373r0, SSC-3: Tape end-to-end data protection(Presentation) as the introduction to this proposal.

IBM Tape has been required to address the question of why our tape drives do not support the T10 standard end-to-end data protection that is available for disk drives. While we have been able to show that it is a disk drive centric solution and that it does not work for tapes, we then have to address the issue of this being one more reason disk should replace tape. That is, disk is viewed as more reliable than tape. We believe that because disk devices have a standard end-to-end data protection they are given an additional marketing tool to win over a tape solution.

IBM believes that SSC-3 needs to provide an end-to-end data protection for tape devices. IBM enterprise drives have been using a proprietary method of end-to-end data protection for over 12 years and we believe that this concept can be easily adapted to fit into a standard.

This end-to-end data protection is accomplished by adding a 32 bit CRC to each data block at the host and transferring that CRC along with the data and validating and storing that CRC with the data on media.

3. Proposal

EDITOR'S NOTE: All new sections.

4.2.23 End-to-end data protection

4.2.23.1 End-to-end data protection overview

A device compliant with this standard may contain hardware or software that is capable of checking and/or generating protection information that is transferred with data and/or commands between the device server and an application client. This protection information is saved to media with each logical block and read from media with each logical block. The configuration of this capability is performed using the Control Data Protection mode page (see 8.3.9). A device that supports using this protection information shall support the Control Data Protection mode page.

4.2.23.2 Protecting logical blocks transferred during reads

A device that is configured to add the protection information to each logical blocks transferred during reads (see 8.3.9) shall read the protection information from the medium and send it with the logical block to the application client. The protection information shall be validated before sending status to the command that caused the transfer of the logical block. The device may also validate the protection information at vendor-specific points in the device. If the validation of the protection information fails, the device server shall respond as defined in the Control Data Protection mode page. A device that supports using the logical block protection information shall support using the protection information during reads.

An application client that has set up the device server to add protection information to each logical block transferred during reads should validate the protection information on each logical block at the latest point possible before using the data. An application client may also validate the data at other points.

4.2.23.3 Protecting logical blocks transferred during writes

A device that is configured to receive the protection information with logical blocks transferred during writes (see 8.3.9) shall save the protection information to medium with the logical block. The protection information shall be validated before the logical block is written to medium and should be validated at the latest possible point. The device may also validate the protection information at other vendor-specific points in the device. If the validation of the protection information fails, the device server shall respond as defined in the Control Data Protection mode page. When in use, this protection information shall be validated before sending status to the command that caused the transfer of the logical block. A device that supports using the logical block protection information shall support using the protection information during writes.

An application client that has set up the device server to add protection information to each logical block transferred during writes should add the protection information on each logical block before transferring that data. The application client should add the protection information to the

data at the earliest point possible. If the data has had the protection information added to the logical block at some point in the application client prior to the hardware that transfers the logical block, the protection information should be validated when it is transferred. If the validation fails, the application client should abort the command and report a status to the user that validation failed.

What is appropriate to do here regarding the inner workings of the application client? Can I specify this and is this the correct way to specify it?

4.2.23.4 Protecting data transferred to the Data-In Buffer

A device that is configured to add the protection information to parameter data transferred from the Data-In Buffer (see 8.3.9) shall generate the protection information and send it with the logical block to the application client. The protection information should be validated before sending status to the command that caused the transfer of the data if the protection information was generated by a different process than the process that transferes the data. The device may also validate the protection information at vendor-specific points in the device. If the validation of the protection information fails, the device server shall respond as defined in the Control Data Protection mode page. A device that supports using protection information may support using the protection information for data transferred to the Data-In Buffer.

An application client that has set up the device server to add protection information to data transferred to the Data-In Buffer should validate the protection information on the data at the latest point possible before using the data. An application client may also validate the data at other points.

4.2.23.5 Protecting data transferred from the Data-Out Buffer

A device that is configured to receive the protection information with data transferred from the Data-Out Buffer (see 8.3.9) shall validate the protection information prior to using that data. The protection information should be validated at the latest possible point. The device may also validate the protection information at other vendor-specific points in the device. If the validation of the protection information fails, the device server shall respond as defined in the Control Data Protection mode page. When in use, this protection information shall be validated before sending status to the command that caused the transfer of the data. A device that supports using the logical block protection information may support using the protection information on data transferred from the Data-Out Buffer.

An application client that has set up the device server to add protection information to data transferred from the Data-Out Buffer shall add the protection information to the data before transferring that data. The application client should add the protection information to the data at the earliest point possible. If the data has had the protection information added at some point in the application client prior to the hardware that transfers the data, the protection information should be validated when it is transferred. If the validation fails, the application client should abort the command and report a status to the user indicating that the validation failed.

4.2.23.6 Protecting CDBs

A device that is configured to receive the protection information with the CDB (see 8.3.9) shall validate the protection information prior to processing the command. The protection information should be validated at the latest possible point. The device may also validate the protection information at other vendor-specific points in the device. If the validation of the protection information fails, the device server shall respond as defined in the Control Data Protection mode page. A device that supports using the logical block protection information may support using the protection information on CDBs.

An application client that has set up the device server to add protection information to CDBs shall add the protection information to the CDB before sending the command.

4.2.23.7 Protecting data transferred from the object buffer in response to a RECOVER BUFFERED DATA command

A device that is configured to add the protection information to each logical block transferred during a RECOVER BUFFERED DATA command (see 8.3.9) shall:

a) read the protection information from the object buffer if it exists; or

b) generate the protection information if it does not exist

and send it with each logical block to the application client. The protection information for each block shall be validated before sending status to the command. The device may also validate the protection information at vendor-specific points in the device. If the validation of the protection information fails for any logical block, the device server shall terminate the command without transferring any additional logical blocks that may exist in the object buffer and respond as defined in the Control Data Protection mode page. A device that supports using the logical block protection information shall support using the protection information during data transfers in resonse to a RECOVER BUFFERED DATA command.

An application client that has set up the device server to add protection information to each logical block transferred during a RECOVER BUFFERED DATA command should validate the protection information on each logical block at the latest point possible before using the data. An application client may also validate the data at other points.

8.3.9 Control Data Protection mode page

The Control Data Protection mode page provides controls that allow selective use of end-to-end data protection. The mode page policy of this page shall be Per I_T nexus.

TABLE x1. Control Data Protection mode page format

Byte	Bit							
	7	6	5	4	3	2	1	0
0	PS	SPF(1b)	PAGE CODE (0Ah)					
1	SUBPAGE CODE (80h) EDITOR'S NOTE: Subpages for the Control (0Ah) mode page are not defined except for subpage 01h. Ideally, we could get CAP to specify that subpages 08h - FEh of page code 0Ah are device-type specific. Then we can specify this page. This seems to be a control setting and as such belongs as a subpage to 0Ah.							
2	(MSB)	PAGE LENGTH (n-3)						
3								(LSB)
4	END-TO-END LOGICAL BLOCK PROTECTION METHOD							
5	BPP		LOGICAL BLOCK PROTECTION INFORMATION LENGTH					
6	RDP	WDP	PDIP	PDOP	CDBP	RBDP	Reserved	
7	CDBTL	PIE	Reserved					
8	RDPR		WDPR		PDIPR		PDOPR	
9	CDBPR		RBDPR		Reserved			
10	Reserved							
n	EDITOR'S NOTE: I wish to leave reserved bytes - at least show that there may be some additional bytes in the future - in case anybody ever decides to do anything on a basis larger than logical block (e.g., file) for end-to-end protection. This seems like a potential extension - although a difficult one.							

QUESTION: Should we add the option to pad the protection information. In the only defined protection information a CRC is used. Currently the CRC is added on a byte boundary (i.e., immediately with no padding). If we add a pad to align to a 4-byte boundary, how do we indicate in the data stream the size of the pad? This would add great complexity and for this reason, I do not want to do it.

The FC_CRC bit set to one indicates that the device server supports using end-to-end data protection using the 4-byte Fibre Channel CRC (see FC-FS-2). The FC_CRC bit set to zero indicates the device server does not support end-to-end data protection using the 4-byte Fibre Channel CRC.

The RS_CRC bit set to one indicates that the device server supports using end-to-end data protection using a 4-byte Reed-Solomon CRC (see ECMA-319). The RS_CRC bit set to zero indicates that the device server does not support end-to-end data protection using a 4-byte Reed-Solomon CRC.

The END-TO-END LOGICAL BLOCK PROTECTION METHOD is defined in Table x2.

TABLE X2. END-TO-END DATA PROTECTION METHOD values

Value	Description
00h	Do not use end-to-end logical block protection
01h	Use the Reed-Solomon CRC (see ECMA-319) as the end-to-end logical block protection information.
02h	Use the 4-byte Fibre Channel CRC (see FC-FS-2) as the end-to-end logical block protection information.
03h - FFh	Reserved

The block protection placement (BPP) field is defined in Table x3.

TABLE x3. Block protection placement values

Value	Definition
00b	The logical block protection information is appended to the data
01b	The logical block protection information is prepended to the data
10b - 11b	Reserved

The LOGICAL BLOCK PROTECTION INFORMATION LENGTH specifies the length of the logical block protection information.

The read data protected (RDP) bit set to one indicates that the protection information is included with data transferred when reading. The RDP bit set to zero indicates that the protection information is not included with data transferred when reading.

The write data protected (WDP) bit set to one indicates that the protection information is included with data transferred when writing. The WDP bit set to zero indicates that the protection information is not included with data transferred when writing.

The parameter data in protected (PDIP) bit set to one indicates that the protection information is included with the data transferred with parameter data in commands. The PDIP bit set to zero indicates that the protection information is not included with the data transferred with parameter data in commands.

The parameter data out protected (PDOP) bit set to one indicates that the protection information is included with the data transferred with parameter data out commands. The PDOP bit set to zero indicates that the protection information is not included with the data transferred with parameter data out commands.

The command descriptor block protected (CDBP) bit set to one indicates that the protection information is included with the CDB when it is transferred. The CDBP bit set to zero indicates that the protection information is not included with the CDB when it is transferred.

The recover buffered data protected (RBDP) bit set to one indicates that the protection information is transferred with the data transferred by the RECOVER BUFFERED DATA command. The RBDP bit set to zero indicates that the protection information is not transferred with the data transferred by the RECOVER BUFFERED DATA command.

The command descriptor block transfer length (CDBTL) bit set to one indicates that the transfer length includes the protection information. The CDBTL bit set to zero indicates that the transfer length does not include the protection information.

The protection information endian (PIE) bit set to one indicates that the protection information is Big Endian. The PIE bit set to zero indicates that the protection information is Little Endian.

The read data protection reporting (RDPR) information is defined in Table x4

TABLE x4. RDPR definition

Value	Device server behavior when the validation of the data fails
00b	Report a Check Condition using a Sense Code of Current Sense and the additional sense code set to END-TO-END LOGICAL BLOCK PROTECTION ERROR ON READ. EDITOR'S NOTE: END-TO-END LOGICAL BLOCK PROTECTION ERROR ON READ is a new additional sense code
01b	Establish a Check Condition for return on the next eligible command with the Sense Code set to Deferred Sense and the additional sense code set to END-TO-END LOGICAL BLOCK PROTECTION ERROR ON READ.
10b - 11b	Reserved

The write data protection reporting (WDPR) field is defined in Table x5

TABLE x5. WDPR definition

Value	Device server behavior when the validation of the data fails
00b	Report a Check Condition using a Sense Code of Current Sense and the additional sense code set to END-TO-END LOGICAL BLOCK PROTECTION ERROR ON WRITE. EDITOR'S NOTE: END-TO-END LOGICAL BLOCK PROTECTION ERROR ON WRITE is a new additional sense code
01b	Establish a Check Condition for return on the next eligible command with the Sense Code set to Deferred Sense and the additional sense code set to END-TO-END LOGICAL BLOCK PROTECTION ERROR ON WRITE.
10b - 11b	Reserved

The parameter data in protected reporting (PDIPR) field is defined in Table x6

TABLE x6. PDIPR definition

Value	Device server behavior when the validation of the data fails
00b	Report a Check Condition using a Sense Code of Current Sense and the additional sense code set to END-TO-END PROTECTION ERROR ON PARAMETER DATA IN. EDITOR'S NOTE: END-TO-END PROTECTION ERROR ON PARAMETER DATA IN is a new additional sense code
01b	Establish a Check Condition for return on the next eligible command with the Sense Code set to Deferred Sense and the additional sense code set to END-TO-END PROTECTION ERROR ON PARAMETER DATA IN. QUESTION: This seems like an overkill and like there would not be a need for this option. I have it here for consistency, but I wonder if it should be removed?
10b - 11b	Reserved

The parameter data out protected reporting (PDOPR) field is defined in Table x7

TABLE x7. PDOPR definition

Value	Device server behavior when the validation of the data fails
00b	Report a Check Condition using a Sense Code of Current Sense and the additional sense code set to END-TO-END PROTECTION ERROR ON PARAMETER DATA OUT. EDITOR'S NOTE: END-TO-END PROTECTION ERROR ON PARAMETER DATA OUT is a new additional sense code
01b	Establish a Check Condition for return on the next eligible command with the Sense Code set to Deferred Sense and the additional sense code set to END-TO-END PROTECTION ERROR ON PARAMETER DATA OUT. QUESTION: This seems like an overkill and like there would not be a need for this option. I have it here for consistency, but I wonder if it should be removed?
10b - 11b	Reserved

The command descriptor block protected reporting (CDBPR) field is defined in Table x8

TABLE x8. CDBPR definition

Value	Device server behavior when the validation of the data fails
00b	Report a Check Condition using a Sense Code of Current Sense and the additional sense code set to END-TO-END PROTECTION ERROR ON CDB. EDITOR'S NOTE: END-TO-END PROTECTION ERROR ON CDB is a new additional sense code
01b	Establish a Check Condition for return on the next eligible command with the Sense Code set to Deferred Sense and the additional sense code set to END-TO-END PROTECTION ERROR ON CDB. QUESTION: This seems like an overkill and like there would not be a need for this option. I have it here for consistency, but I wonder if it should be removed?
10b - 11b	Reserved

The recover buffered data protected reporting (RBDPR) field is defined in Table x9

TABLE x9. RBDPR definition

Value	Device server behavior when the validation of the data fails
00b	Report a Check Condition using a Sense Code of Current Sense and the additional sense code set to END-TO-END LOGICAL BLOCK PROTECTION ERROR ON RECOVER BUFFERED DATA. EDITOR'S NOTE: END-TO-END LOGICAL BLOCK PROTECTION ERROR ON RECOVER BUFFERED DATA is a new additional sense code
01b	Establish a Check Condition for return on the next eligible command with the Sense Code set to Deferred Sense and the additional sense code set to END-TO-END LOGICAL BLOCK PROTECTION ERROR ON RECOVER BUFFERED DATA.
10b - 11b	Reserved