To: **T10 Committee**
From: **Gerry Houlder, Seagate Technology,** gerry_houlder@seagate.com
Developed for **Trusted Computing Group**, www.trustedcomputinggroup.org
Subj: **SPC-4 Security Commands proposal**
Date: **Oct. 7, 2005**

This document presents a proposal for defining an industry standard set of interface commands for a trusted device, which is a component of an overall trusted system.

A trusted device provides a horizontal security product embedded in devices whose behavior may be authorized via interaction with a trusted host system.

This proposal uses two commands: TRUSTED OUT and TRUSTED IN. These commands provide for variable length data transfers. These commands are 12 byte CDBs to provide portability between SCSI and ATAPI implementations.

The CDB parameters shall be defined by T10. The data payload and subsequent actions resulting from these commands are defined by the Trusted Protocol identified in the CDB. The intent is to standardize this data content so it is identical across both ATA and SCSI.

2.2 Approved References

- ITU-T RECOMMENDATION X.509 | ISO/IEC 9594-8, *Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks*, ITU, 2000.
- Information processing systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1), International Organization for Standardization. International Standard 8824, (December, 1987).

2.4 IETF References

- RFC 3280, *Internet X.509 Public Key Infrastructure: Certificate and Certificate Revocation List (CRL) Profile*, IETF, 2002.
- RFC 3281, *An Internet Attribute Certificate: Profile for Authorization*, IETF, 2002.

3.2 Abbreviations

3.2.a TCG: Trusted Computing Group. Web site at https://www.trustedcomputinggroup.org.

3.2.b OID: Object Identifier. See ASN.1 and ISO/IEC 9834/ ITU-T X.622.

4.3.4.4 Transfer length
[Note: No changes are proposed – included for reference only.]

The TRANSFER LENGTH field specifies the amount of data to be transferred, usually the number of blocks. Some commands use transfer length to specify the requested number of bytes to be sent as defined in the command description.

Commands that use one byte for the TRANSFER LENGTH field may allow up to 256 blocks or 256 bytes of data to be transferred by one command.

In commands that use multiple bytes for the TRANSFER LENGTH field, a transfer length of zero specifies that no data transfer shall take place. A value of one or greater specifies the number of blocks or bytes that shall be transferred.

Refer to the specific command description for further information.

4.3.4.6 Allocation length
[Note: For this clause, changes from current wording are underlined.]

The ALLOCATION LENGTH field specifies the maximum number of bytes <u>or blocks</u> that an application client has allocated in the Data-In Buffer. <u>The field specifies bytes unless defined differently by the command.</u>

An allocation length of zero specifies that no data shall be transferred. This condition shall not be considered as an error.

The device server shall terminate transfers to the Data-In Buffer when the number of bytes specified by the ALLOCATION LENGTH field have been transferred or when all available data have been transferred, whichever is less. The allocation length is used to limit the maximum amount of variable length data (e.g., mode data, log data, diagnostic data) returned to an application client. If the information being transferred to the Data-In Buffer includes fields containing counts of the number of bytes in some or all of the data, then the contents of these fields shall not be altered to

reflect the truncation, if any, that results from an insufficient ALLOCATION LENGTH value, unless the standard that describes the Data-In Buffer format states otherwise.

If the amount of information to be transferred exceeds the maximum value that the ALLOCATION LENGTH field is capable of specifying, the device server shall transfer no data and 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.

5.6.1 Persistent Reservations overview

[In table 31, add the TRUSTED IN and TRUSTED OUT commands as “allowed” for all 5 columns. The underlaying protocol is expected to provide its own protection against interference from other initiators and this will eliminate reservations as a source of “denial of service” attacks.]

6.x Trusted Out command

The TRUSTED OUT command (see table 1) is used to send data to the device server. The data sent contains one or more TRUSTED PROTOCOL specific instructions to be performed by the device server. The application client uses TRUSTED IN command to retrieve data derived from these instructions.

Table 1 – Trusted Out command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (B5h)							
1	TRUSTED PROTOCOL							
2	TP_SPECIFIC							
3								
4	RESERVED							
5								
6								
7								
8	(MSB)	TRANSFER LENGTH						
9								(LSB)
10	RESERVED							
11	CONTROL							

The TRUSTED PROTOCOL field specifies which trusted protocol is being used. This determines the format of the data that is sent (see table 2).

Table 2 – TRUSTED OUT TRUSTED PROTOCOL field

Code	Description
00h	Reserved
01h – 06h	Defined by the TCG (see 3.2)
07h – EFh	Reserved
F0h – FFh	Vendor specific

The TP_SPECIFIC field provides TRUSTED PROTOCOL field specific information. The meaning of this field is defined by each trusted protocol.

The TRANSFER LENGTH field specifies the number of bytes to be sent and is expressed in increments of 512 bytes (e.g., a value of one means 512 bytes, two means 1 024 bytes, etc.). Pad bytes are appended as needed to meet this requirement. Pad bytes shall have a value of 00h.

The device server shall return GOOD status as soon as it determines the data has been correctly received. This does not indicate that the data has been parsed or processed. These indications are only obtained by sending a TRUSTED IN command and receiving the results in the associated data transfer.

The format of the data depends on the protocol specified by the TRUSTED PROTOCOL field (see table 2).

6.y Trusted In command

6.y.1 Trusted In command description

The TRUSTED IN command (see table 3) is used to retrieve trusted protocol information (see 6.y.2) or the results of one or more TRUSTED OUT commands (see 6.x).

Table 3 – Trusted In command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (A2h)							
1	TRUSTED PROTOCOL							
2	TP_SPECIFIC							
3								
4	RESERVED							
5								
6								
7								
8	(MSB)	ALLOCATION LENGTH						
9								(LSB)
10	RESERVED							
11	CONTROL							

The TRUSTED PROTOCOL field specifies which trusted protocol is being used. This determines the format of the data that is transferred (see table 4).

Table 4 – TRUSTED IN TRUSTED PROTOCOL field

Code	Description
00h	Return trusted protocol information (see 6.y.2)
01h – 06h	Defined by the TCG (see 3.2)
07h – EFh	Reserved
F0h - FFh	Vendor specific

The TP_SPECIFIC field provides TRUSTED PROTOCOL field specific information. The meaning of this field is defined by each trusted protocol.

The ALLOCATION LENGTH field specifies the maximum number of bytes available to receive data and is expressed in increments of 512 bytes (e.g., a value of one means 512 bytes, two means 1 024 bytes, etc.). Pad bytes are appended to the next 512 byte boundary as needed to meet this requirement. Pad bytes shall have a value of 00h. If the allocation length is not sufficient to return all of the data bytes the device server has available to transfer, the device server shall transfer as many bytes as possible consistent with the requirements of the specified TRUSTED PROTOCOL field without exceeding the allocation length and the command shall be completed with GOOD status. Indications of data overrun or underrun or the mechanism, if any, for processing retries are TRUSTED PROTOCOL field specific.

Any linkage between a previous TRUSTED OUT command and the data returned by a TRUSTED IN command is TRUSTED PROTOCOL field specific. If the device server has no data to transfer (e.g., the results for any previous TRUSTED OUT commands are not yet available), the device server may return TRUSTED PROTOCOL field specific data indicating it has no other data to return. The command shall be completed with GOOD status unless a transport protocol failure (e.g., parity or CRC error is detected) occurs.

For TRUSTED PROTOCOL field set to 00h, the format for the data is described in 6.y.2. The format of the data for other TRUSTED PROTOCOL values is documented by the group that owns the associated TRUSTED PROTOCOL value.

The device server shall retain data resulting from a TRUSTED OUT command awaiting retrieval by a TRUSTED IN command until one of the following events is processed:

a) the data is delivered according to the TRUSTED PROTOCOL field (see table 4) specific rules for the TRUSTED IN command;
b) logical unit reset; or
c) I_T nexus loss associated with the I_T nexus that sent the TRUSTED OUT command.

If the data is lost due to one of these events and the application client still wants to perform the TRUSTED PROTOCOL specific instruction, the application client may send a new TRUSTED OUT command.

6.y.2 TRUSTED PROTOCOL 00h description

6.y.2.1 CDB description

The purpose of TRUSTED PROTOCOL of 00h is to return trusted protocol related information for the logical unit. A TRUSTED IN command using TRUSTED PROTOCOL field set to 00h is not linked to an earlier TRUSTED OUT command. When the TRUSTED PROTOCOL field is set to 00h, the valid TP_SPECIFIC field options are shown in table 5.

Table 5 – TP_SPECIFIC field

Code	Description
0000h	Return a certificate (see 6.y.2.2)
0001h	Return supported trusted protocol list (see 6.y.2.3)
0002h – FFFFh	Reserved

If the TP_SPECIFIC field is set to a reserved value, the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST and the additional sense code

set to INVALID FIELD IN CDB. All other CDB fields for TRUSTED IN command shall meet the requirements of 6.y.1.

Each time a TRUSTED IN command with TRUSTED PROTOCOL field set to 00h is received, the device server shall transfer the bytes starting with byte 0.

6.y.2.2 Certificate data description

6.y.2.2.1 Certificate overview

A certificate is either an X.509 Attribute Certificate (see 6.y.2.2.3) or an X.509 Public Key Certificate (see 6.y.2.2.2) depending on the capabilities of the logical unit.

When the TRUSTED PROTOCOL field is set to 00h and the TP_SPECIFIC field is set to 0000h in a TRUSTED IN command, the parameter data shall have the format shown in Table 7.

Table 7 – TRUSTED IN parameter data for TP_SPECIFIC 0000h

Bit / Byte	7	6	5	4	3	2	1	0
0	RESERVED							
1	RESERVED							
2	(MSB)	CERTIFICATE LENGTH (m - 3)						
3								(LSB)
4	CERTIFICATE							
m								
m+1	PAD BYTES (if any)							
n								

The CERTIFICATE LENGTH field indicates the total length, in bytes, of the certificate. This length includes one or more certificates. If the device server doesn't have a certificate to return, the certificate length shall be set to 0000h and only the 4 byte header followed by 508 pad bytes shall be available for transfer.

The contents of the certificate fields are defined in 6.y.2.2.2 and 6.y.2.2.3.

The total data length shall conform to the ALLOCATION LENGTH field requirements (i.e., the total data length shall be a multiple of 512). Pad bytes are appended as needed to meet this requirement. Pad bytes shall have a value of 00h.

6.y.2.2.2 Public Key certificate description

RFC 3280 defines the certificate syntax for certificates consistent with X.509v3 Public Key Certificate Specification. Table 8 describes the trusted command usage of the X.509 public key certificate fields and the relationship of that usage to the definitions of RFC 3280.

Table 8 –Usage of X.509 certificate values in RFC 3280 context

Certificate Field [1]	Details
SignatureAlgorithm	As described in RFC 3280
SignatureValue	As described in RFC 3280
Version	Shall be set to 2 (i.e., version 3)
SerialNumber	As described in RFC 3280
Signature	As described in RFC 3280
Issuer	As described in RFC 3280 with the added constraint that UTF8String encoding of DirectoryString shall be used.
Validity	As described in RFC 3280. The Begin Date should be set to the time of credential issuance. To indicate no expiration date, the Expiration Date should be set to the Begin Date plus 100 years.
Subject	As described in RFC 3280. Information contained in this field shall either be populated with a non-empty distinguished name identifying the device or a null value.
SubjectPublicKeyInfo	As described in RFC 3280
subject Alternate Name Extension	As described in RFC 3280, but may be ignored. This standard restricts the use to the following options only: a) otherName; or b) directoryName. subjectAltName shall contain only one of the following: a) The device serial number using directoryName; or b) The device serial number using otherName. If this field is used then subject field shall contain a null value.
basicConstraints Extension	As described in RFC 3280
cRLDistributionPoints Extension	As described in RFC 3280
subjectDirectoryAttributes Extension: protocols	Sequence of OIDs
[1] Certificate field names are as described in RFC 3280.	

6.y.2.2.3 Attribute certificate description

RFC 3281 defines the certificate syntax for certificates consistent with X.509v2 Attribute Certificate Specification. Table 9 describes the trusted command usage of the X.509 attribute key certificate fields and the relationship of that usage to the definitions of RFC 3281.

Table 9 –Usage of X.509 certificate values in RFC 3281 context

Certificate Field [1]	Details
SignatureAlgorithm	As described in RFC 3281
SignatureValue	As described in RFC 3281
Version	Shall be set to 1 (i.e., version 2)
Holder	As described in RFC 3281 with the added constraint that entityName option shall contain one of the of the following values: a) an URI using uniformResourceIdentifier; b) the device serial number using directoryName or otherName; or c) a null value.
issuer	As described in RFC 3281
signature	As described in RFC 3281
serialNumber	As described in RFC 3281
attrCertValidityPeriod	As described in RFC 3281. The Begin Date should be set to the time of credential issuance. To indicate no expiration date, the Expiration Date should be set to the Begin Date plus 100 years.
attributes: protocols	Sequence of OIDs
basicAttConstraints Extension	As described in RFC 3281
cRLDistributionPoints Extension	As described in RFC 3281
[1] Certificate field names are as described in RFC 3281.	

6.y.2.3 Supported trusted protocols list description

When the TRUSTED PROTOCOL field is set to 00h and the TP_SPECIFIC field is set to 0001h in a TRUSTED IN command, the parameter data shall have the format shown in Table 10.

Table 10 – TRUSTED IN parameter data for TP_SPECIFIC 0001h

Bit / Byte	7	6	5	4	3	2	1	0
0	RESERVED							
1	RESERVED							
2	(MSB)	LIST LENGTH (m - 3)						
3								(LSB)
4	SUPPORTED TRUSTED PROTOCOL LIST							
m								
m+1	PAD BYTES (if any)							
n								

The LIST LENGTH field indicates the total length, in bytes, of the supported trusted protocol list.

The SUPPORTED TRUSTED PROTOCOL LIST field shall contain a list of all supported TRUSTED PROTOCOL field values. Each byte indicates a supported TRUSTED PROTOCOL field value. The values shall be in ascending order starting with 00h.

The total data length shall conform to the ALLOCATION LENGTH field requirements (i.e., the total data length shall be a multiple of 512). Pad bytes are appended as needed to meet this requirement. Pad bytes shall have a value of 00h.