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: **Sept. 1, 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 Security Protocol identified in the CDB. The intent is to standardize this data content so it is identical across both ATA and SCSI.

0.1 Reference documents

- 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.
- ITU-T RECOMMENDATION X.509 | ISO/IEC 9594-8, *Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks*, ITU, 2000.

0.2 Definitions

0.2.a Trusted Action: A group of bytes that describe a security procedure that a device server is requested to perform.

0.2.b Trusted Result: A group of bytes that contain the results of a requested trusted action or the status from processing a requested trusted action.

0.3 Abbreviations

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

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 actions to be performed by the device server. The application client uses TRUSTED IN command to retrieve trusted results derived from the trusted actions.

Table 1 – Trusted Out command

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

The SECURITY PROTOCOL IDENTIFICATION field identifies which security protocol is being used. This determines the format of the data that is transferred. The meaning of the SECURITY PROTOCOL IDENTIFICATION value is described in table 2.

Table 2 – Security Protocol Identification field

Code	Description
00h	Reserved
01h – 06h	Defined to TCG
07h – EFh	Reserved
F0h – FFh	Vendor specific

The TRANSFER LENGTH field specifies the number of bytes to be transferred and is expressed in increments of 512 bytes (i.e., 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 that any trusted actions have been 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 is dependent on the protocol specified by the SECURITY PROTOCOL IDENTIFICATION 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 results.

Table 3 – Trusted In command

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

The SECURITY PROTOCOL IDENTIFICATION field identifies which security protocol is being used. This determines the format of the data that is transferred. The meaning of the SECURITY PROTOCOL IDENTIFICATION value is defined in table 4.

Table 4 – Security Protocol Identification field

Code	Description
00h	Return X3.509 certificate
01h – 06h	Defined to TCG
07h – EFh	Reserved
F0h - FFh	Vendor specific

The ALLOCATION LENGTH field specifies the maximum number of bytes to be transferred and is expressed in increments of 512 bytes (i.e., 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 send, the device server shall send as many complete trusted results as possible without exceeding the transfer length and the command shall end with GOOD status. Indications of data overrun or underrun or any needed retries are SECURITY PROTOCOL IDENTIFICATION field specific.

Any linkage between a previous TRUSTED OUT command and the trusted results returned by a TRUSTED IN command is SECURITY PROTOCOL IDENTIFICATION field specific. If the device server has no trusted results to send (e.g., trusted results for a previously requested trusted action are not ready yet), the device server may return a SECURITY PROTOCOL IDENTIFICATION field specific trusted result indicating it has no data to return and the command shall end with GOOD status.

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

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

a) a matching TRUSTED IN command;
b) logical unit reset;
c) I_T nexus loss associated with the I_T nexus that sent the TRUSTED OUT command;
d) processing any of the following task management functions (see SAM-3):
    A) CLEAR TASK SET;
    B) ABORT TASK SET.

If the trusted result data is lost due to one of these events and the application client still wants to perform the trusted action, the application client is required to resend the trusted action in a new TRUSTED OUT command.

6.y.2 Security Protocol ID 00h description

6.y.2.1 CDB description

The purpose of SECURITY PROTOCOL IDENTIFICATION of 00h is to return the X3.509 certificate for the logical unit. Typically this security identification credential is retrieved as part of a discovery process before initiating a specific security protocol with the logical unit.

Note: X.509 certificates are designed to be transferred and/or stored as plaintext.

For SECURITY PROTOCOL IDENTIFICATION field set to 00h, the SP_SPECIFIC bytes are defined as shown in table 5.

Table 5 – SP_SPECIFIC description for SECURITY PROTOCOL IDENTIFICATION field = 00h

Bit / Byte	7	6	5	4	3	2	1	0
1	SECURITY PROTOCOL IDENTIFICATION = 00H							
2	SP_SPECIFIC							
3								

The SP_SPECIFIC field shall be set to 0000h for TRUSTED IN command. All other CDB fields for TRUSTED IN command shall meet the requirements of 6.y.1.

The format of the returned data is described in 6.y.2.2. Each time a TRUSTED IN command with SECURITY PROTOCOL IDENTIFICATION field set to 00h is received, the device server shall transfer the bytes starting with byte 0.

6.y.2.2 Certificate header

When the SECURITY PROTOCOL IDENTIFICATION field is set to 00h in a TRUSTED IN command, the header, the certificate bytes, and pad bytes (if any) shall be returned as shown in Table 7.

Table 7 – X.509 header and certificate description

Bit / Byte	7	6	5	4	3	2	1	0
0	RESERVED							
1	RESERVED							
2	(MSB)	CERTIFICATE LENGTH (m - 3)						
3								(LSB)
4	X.509 CERTIFICATE BYTES							
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 is set to 0000h and only the 4 byte header and 508 pad bytes are returned.

The total data length shall conform to the ALLOCATION LENGTH field requirements (e.g., 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.2.3 Certificate description

6.y.2.3.1 Certificate overview

The instantiation of a X.509 conformant credential is either through an X.509 Attribute Certificate or an X.509 Public Key Certificate depending on the capabilities of the logical unit.

6.y.2.3.2 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 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. It is recommended to set Begin Date to the time of credential issuance and the Expiration Date to the Begin Date plus one hundred years if the intent is not to indicate an expiration date.
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 specification restricts the use to the following options only: • otherName; • directoryName. One and only one of the following values is allowed for subjectAltName: • The device serial number using directoryName; • 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 OID Defines supported Security and Integrity Protocols
[1] Certificate field names are as described in RFC 3280.	

6.y.2.3.3 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 version 2.
Holder	As described in RFC 3281 with the added constraint that entityName option be used in the Holder field containing one and only one of the of the following values: • an URI using uniformResourceIdentifier; • the device serial number using directoryName or otherName; • 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. It is recommended to set Begin Date to the time of credential issuance and the Expiration Date to the Begin Date plus one hundred years if the intent is not to indicate an expiration date.
attributes: protocols	SEQUENCE OF OID Defines supported Security and Integrity Protocols.
basicAttConstraints Extension	As described in RFC 3281.
cRLDistributionPoints Extension	As described in RFC 3281.
[1] Certificate field names are as described in RFC 3281.	