To: **T10 Committee**
From: **Gerry Houlder, Seagate Technology,** gerry_houlder@seagate.com
Developed for **Trusted Computing Group**, www.trustedcomputinggroup.org
Subj: **SPC-3 Security Commands proposal**
Date: **April 29, 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. This proposal refers to the data payload format as “restricted” to indicate that the format shall conform to the definition established by the outside group that “owns” the particular SPID (Security Protocol ID) value.

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.

1.1 Trusted Out command

The TRUSTED OUT command (see table 1) is used to send trusted data to the device server. The data sent contains one or more trusted actions to be performed by the device server. The application client shall use TRUSTED IN command to retrieve any 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	RESERVED			RSTRICT	SPID			
2	PKT_TYP	SP_CTRL		RSTRICT	IPID			
3	RESTRICTED							
4	(MSB)	KEY REFERENCE						
5								
6								
7								(LSB)
8	(MSB)	TRANSFER LENGTH						
9								(LSB)
10	RESERVED							
11	CONTROL							

The SPID (Security Protocol Identification) field identifies which security protocol is being used. This determines the format of the data that will be transferred. The meaning of the SPID values is described in table 2.

Table 2 – Security Protocol ID field description

Value	Description
0h	Request for an X.509 certificate (see 1.3)
1h – 6h	Reserved.
7h – Ch	Restricted to TCG.
Dh - Fh	Vendor specific.

A PKT_TYP (packet type) bit set to one indicates that multiple trusted actions may be sent in the payload. A PKT_TYP bit set to zero indicates that a single trusted action shall be sent.

The SP_CTRL (Security Protocol Control) field provides security protocol specific control information. The meaning of these bits is defined by each security protocol.

The IPID (Integrity Protocol Identification) field identifies which integrity protocol is being used to protect the data. The meaning of the IPID values is described in table 3.

Table 3 – Integrity Protocol ID field description

Value	Description
0h	No integrity protection.
1h	Hashed.
2h	Keyed hashed (HMAC).
3h	SHA-1.
4h	SHA-256.
5h - Dh	Reserved.
Eh - Fh	Vendor specific.

The KEY REFERENCE field provides a pointer to the key used with the integrity protocol to validate the data.

The TRANSFER LENGTH field specifies the number of blocks of data to be transferred. The block size for this command shall be 512 bytes regardless of the actual logical block size reported by the READ CAPACITY command.

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 data is organized as one or more trusted actions. For SPID set to zero, The TRANSFER LENGTH shall be zero and no data is transferred. In this case, if the transfer length is non-zero the command is terminated with CHECK CONDITION status with sense key set to ILLEGAL REQUEST and ASC set to Illegal Bit or Byte in CDB. The format of the trusted actions for other SPID values is restricted. The format is documented by the group that owns the associated SPID value (e.g., the data format for a value of 7h is documented by TCG).

1.2 Trusted In command

The TRUSTED IN command (see table 4) is used to retrieve trusted results from trusted actions that were sent in a previous TRUSTED OUT command.

Table 4 – Trusted In command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (A2h)							
1	RESERVED			RSTRICT	SPID			
2	PKT_TYP	SP_CTRL		RSTRICT	IPID			
3	RESTRICTED							
4	(MSB)	KEY REFERENCE						
5								
6								
7								(LSB)
8	(MSB)	TRANSFER LENGTH						
9								(LSB)
10	RESERVED							
11	CONTROL							

The SPID (Security Protocol Identification) field identifies which security protocol is being used. This determines the format of the data that will be transferred. The meaning of the SPID values is defined in table 2. The device server shall only return trusted results that match the SPID value requested.

A PKT_TYP (packet type) bit set to one indicates that multiple trusted results may be returned by the device server. A PKT_TYP bit set to zero indicates that only one trusted result shall be returned.

The SP_CTRL (Security Protocol Control) field provides security protocol specific control information. The meaning of these bits is defined by each security protocol.

The IPID (Integrity Protocol Identification) field identifies which integrity protocol is being used to protect the data. The meaning of the IPID values is described in table 3. The device server shall only return trusted results that match the IPID value requested.

The KEY REFERENCE field provides a pointer to the key used with the integrity protocol to validate the data.

The TRANSFER LENGTH field is described in xxx. [Editor's note: reference to standard paragraph defining Transfer Length] The block size for this command shall be 512 bytes regardless of the actual logical block size reported by the READ CAPACITY command. If the length is not sufficient to return all of the blocks the device server has available to send, the device server shall send as many complete trusted results as possible without exceeding the transfer length.

If the device server has no trusted results to send, the device server shall return a trusted result indicating it has no data to return and the command shall end with GOOD status.

The returned data is organized as one or more trusted results. For SPID set to zero, the returned data is a certificate. The format for this data is described in 1.3. The format of the trusted results for other SPID values is restricted. The format is documented by the group that owns the associated SPID value (e.g., the data format for a value of 7h is documented by TCG).

It is the application client's responsibility to have an outstanding TRUSTED IN command whenever it believes there are trusted results pending in the device server.

[Need rule stating other cases where device server can discard these results. E.g, resets, timeouts, etc.]

1.3 Certificate descriptions

1.3.1 Certificate header

When the SPID field of the TRUSTED IN command specifies an X.509 certificate, a header and the certificate bytes shall be returned as shown in Table 5.

Table 5 – X.509 header and certificate description

Bit / Byte	7	6	5	4	3	2	1	0
0	RESERVED							
1	RESERVED							
2	(MSB)	CERTIFICATE LENGTH						
3								(LSB)
4	X.509 CERTIFICATE BYTES							
N								

The CERTIFICATE LENGTH 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 zero and only the 4 byte header is returned.

1.3.2 Certificate description

1.3.2.1 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 device. A X.509 Attribute Certificate shall be issued for any device not capable of asymmetric key operations or any device for which the credential issuer does not want to include any public key information in the credential. A X.509 Public Key Certificate shall be issued for any device capable of asymmetric key operations and for which the certificate issuer wants to bind the public key to the device.

1.3.2.2 Public Key certificate description

RFC 3280 defines the certificate syntax for certificates consistent with X.509v3 Public Key Certificate Specification. The following extract from RFC 3280 summarizes those parameters.

```
Certificate  ::=  SEQUENCE  {
        tbsCertificate       TBSCertificate,
        signatureAlgorithm   AlgorithmIdentifier,
        signatureValue       BIT STRING  }

   TBSCertificate  ::=  SEQUENCE  {
        version         [0]  EXPLICIT Version DEFAULT v1,
        serialNumber         CertificateSerialNumber,
        signature            AlgorithmIdentifier,
        issuer               Name,
        validity             Validity,
        subject              Name,
        subjectPublicKeyInfo SubjectPublicKeyInfo,
        issuerUniqueID  [1]  IMPLICIT UniqueIdentifier OPTIONAL,
                             -- If present, version MUST be v2 or v3
        subjectUniqueID [2]  IMPLICIT UniqueIdentifier OPTIONAL,
                             -- If present, version MUST be v2 or v3
        extensions      [3]  EXPLICIT Extensions OPTIONAL
                             -- If present, version MUST be v3
        }

   Version  ::=  INTEGER  {  v1(0), v2(1), v3(2)  }

   CertificateSerialNumber  ::=  INTEGER

   Validity ::= SEQUENCE {
        notBefore      Time,
        notAfter       Time }

   Time ::= CHOICE {
        utcTime        UTCTime,
        generalTime    GeneralizedTime }

   UniqueIdentifier  ::=  BIT STRING

   SubjectPublicKeyInfo  ::=  SEQUENCE  {
        algorithm            AlgorithmIdentifier,
        subjectPublicKey     BIT STRING  }

   Extensions  ::=  SEQUENCE SIZE (1..MAX) OF Extension


   Extension  ::=  SEQUENCE  {
        extnID      OBJECT IDENTIFIER,
        critical    BOOLEAN DEFAULT FALSE,
        extnValue   OCTET STRING  }
```

Table 6 describes the SCSI usage of the X.509 public key certificate fields and the relationship of that usage to the definitions of RFC 3280.

Table 6 – SCSI usage of X.509 certificate values in RFC 3280 context

Certificate Field	Usage	Details
signatureAlgorithm	Mandatory	As per RFC 3280.
signatureValue	Mandatory	As per RFC 3280.
version	Mandatory	Shall be version 3.
serialNumber	Mandatory	As per RFC 3280.
signature	Mandatory	As per RFC 3280.
issuer	Mandatory	As per RFC 3280 with the added constraint that UTF8String encoding of DirectoryString shall be used.
validity	Mandatory	As per 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	Mandatory	As per 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	Mandatory	As per RFC 3280.
subject Alternate Name Extension	Supported	As per 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	Supported	As per RFC 3280, but may be ignored.
cRLDistributionPoints Extension	Supported	As per RFC 3280, but may be ignored.
subjectDirectoryAttributes Extension: protocols	Mandatory	SEQUENCE OF OID Defines supported Security and Integrity Protocols

1.3.2.3 Attribute certificate description

RFC 3281 defines the certificate syntax for certificates consistent with X.509v2 Attribute Certificate Specification. Certificates for SCSI may use the RFC 3281 certificate syntax. The following extract from RFC 3281 summarizes those parameters.

```
AttributeCertificate ::= SEQUENCE {
     acinfo               AttributeCertificateInfo,
     signatureAlgorithm   AlgorithmIdentifier,
     signatureValue       BIT STRING  }

AttributeCertificateInfo ::= SEQUENCE {
     version              AttCertVersion -- version is v2,
     holder               Holder,
     issuer               AttCertIssuer,
     signature            AlgorithmIdentifier,
     serialNumber         CertificateSerialNumber,
     attrCertValidityPeriod   AttCertValidityPeriod,
     attributes           SEQUENCE OF Attribute,
     issuerUniqueID       UniqueIdentifier OPTIONAL,
     extensions           Extensions OPTIONAL  }

AttCertVersion ::= INTEGER { v2(1) }

Holder ::= SEQUENCE {
      baseCertificateID   [0] IssuerSerial OPTIONAL,
               -- the issuer and serial number of
               -- the holder's Public Key Certificate
      entityName          [1] GeneralNames OPTIONAL,
               -- the name of the claimant or role
      objectDigestInfo    [2] ObjectDigestInfo OPTIONAL
               -- used to directly authenticate the holder,
               -- for example, an executable  }

ObjectDigestInfo ::= SEQUENCE {
     digestedObjectType  ENUMERATED {
             publicKey            (0),
             publicKeyCert        (1),
             otherObjectTypes     (2) },
                    -- otherObjectTypes MUST NOT
                    -- be used in this profile
     otherObjectTypeID   OBJECT IDENTIFIER OPTIONAL,
     digestAlgorithm     AlgorithmIdentifier,
     objectDigest        BIT STRING  }

AttCertIssuer ::= CHOICE {
     v1Form   GeneralNames,  -- MUST NOT be used in this
                             -- profile
     v2Form   [0] V2Form     -- v2 only  }

V2Form ::= SEQUENCE {
     issuerName            GeneralNames  OPTIONAL,
     baseCertificateID     [0] IssuerSerial  OPTIONAL,
     objectDigestInfo      [1] ObjectDigestInfo  OPTIONAL
       -- issuerName MUST be present in this profile
       -- baseCertificateID and objectDigestInfo MUST NOT
       -- be present in this profile  }

IssuerSerial  ::=  SEQUENCE {
     issuer         GeneralNames,
     serial         CertificateSerialNumber,
     issuerUID      UniqueIdentifier OPTIONAL  }

AttCertValidityPeriod  ::= SEQUENCE {
     notBeforeTime  GeneralizedTime,
     notAfterTime   GeneralizedTime  }
```

Table 7 describes the SCSI usage of the X.509 attribute certificate fields and the relationship of that usage to the definitions of RFC 3281.

Table 7 – SCSI usage of X.509 certificate values in RFC 3281 context

Certificate Field	Usage	Details
signatureAlgorithm	Mandatory	As per RFC 3281.
signatureValue	Mandatory	As per RFC 3281.
version	Mandatory	Shall be version 2.
holder	Mandatory	As per 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	Mandatory	As per RFC 3281.
signature	Mandatory	As per RFC 3281.
serialNumber	Mandatory	As per RFC 3281.
attrCertValidityPeriod	Mandatory	As per 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	Mandatory	SEQUENCE OF OID Defines supported Security and Integrity Protocols.
basicAttConstraints Extension	Supported	As per RFC 3281, but may be ignored.
cRLDistributionPoints Extension	Supported	As per RFC 3281, but may be ignored.