ENDL
TEXAS

Date: 20 April 2008
To: T10 Technical Committee
From: Ralph O. Weber
Subject: Capability-based Command Security (CbCS) [the rewrite]

To Do

- SPIN protocols
- SPOUT protocols
- 5.13.6.8.15 – CbCS parameters
- Glossary
- Acronyms (CbCS only one listed in 07-454r5)
- missing cross references

Introduction

The rush to approve 07-454r5 concurrently with other security proposals such as IKEv2-SCSI and ESP-SCSI has lead to several nettlesome problems. Some prime examples are:

- The ability of the RECEIVE CREDENTIAL command to be used for a plaintext attack on the CbCS SA.
- There is too much dependence on a specific I_T Nexus for SA usage.

Rather than nickel-and-dime these changes, it is proposed to translate 07-454r5 into text that is at least 90% ready for incorporation in SPC-4 (i.e., the content of this proposal), review/revise that, and approve the result as a replacement for the already approved 07-454r5.

Related/Source Documents

This proposal incorporates material from the following T10 proposals:

07-454r5	Capability based Command Security {{not fully incorporated in this revision}}
08-101r1	SPC-4: CbCS field byte alignment changes {{not fully incorporated in this revision}}
08-128r0	SPC-4 RECEIVE CREDENTIAL command 'adjustments'
08-129r0	SPC-4 CbCS capability validation omissions
08-138r0	Constraints on SPC-4 SA creation based on Usage Type
08-141r0	CbCS SECURITY PROTOCOL IN/OUT tweaks in SPC-4

Revision History

r0 Initial revision
r1 Incorporated changes requested by March CAP working group and Sivan Tal. Updated SPC-4 subclause, table, etc. references to SPC-4 r14. Added CbCS page format definitions. Modified the CbCS validation requirements for the SECURITY PROTOCOL IN command to facilitate security token transfers and similar tasks.

Changes between r0 and r1 are indicated by change bars.

Unless otherwise indicated additions are shown in blue, deletions in red strikethrough, and comments in green.

Proposed Changes in SPC-4 r13

3.1.a CbCS extension descriptor: An XCDB descriptor (see 4.3.4.2) in which the EXTENSION TYPE field is set to 40h (i.e., CbCS) as described in 5.13.6.8.16.

4.3.4.2 The XCDB format

…

The EXTENSION TYPE field (see table 12) specifies the size and format of the extension parameters that follow in the XCDB descriptor.

Table 12 — EXTENSION TYPE field

Code	Descriptor Order [a]	Description	Extension size (bytes)	Reference
40h	first	CbCS extension descriptor	140	5.13.6.8.16
all others	Reserved			

[a] The order in which XCDB descriptors appear in an XCDB is arranged so that all the XCDB descriptors that follow an XCDB descriptor defined in a future version of this standard are also XCDB descriptors defined in a future version of this standard (i.e., after encountering one unrecognized XCDB descriptor, all subsequent XCDB descriptors are also going to be unrecognized).

{{The reference to the table footnote should have appeared in SPC-4 r12, but it was not present.}}

…

5.13.2.2 SA parameters

…

The USAGE_TYPE SA parameter shall be one of the values shown in table 49.

Table 49 — USAGE_TYPE SA parameter values

Value [a]	Description	Usage model	Usage data description	Reference
0000h - 0080h	Reserved			
0081h	Tape data encryption	ESP-SCSI [b]	None [c]	SSC-3
0082h - 8000h	Reserved			
8001h	CbCS authentication and credential encryption	ESP-SCSI [b]	None [b]	5.13.6.8
8002h - FFFFh	Reserved			
~~0082h - FFFFh~~	~~Reserved~~			

[a] USAGE_TYPE values between 8000h and CFFFh inclusive place additional constraints on how an SA is to be created as described in 7.6.3.5.13.
[b] ESP-SCSI usage is defined in 5.13.7.
[c] The usage data length field in the IKEv2-SCSI SAUT Cryptographic Algorithms payload (see 7.6.3.5.13) shall contain zero.

…

5.13.6.8 Capability-based command security technique

5.13.6.8.1 Overview

CbCS is a credential-based system that manages access to a logical unit by the coordination between shared keys and security ~~attributes~~ parameters set by the CbCS management application client (see 5.13.6.8.4) and credentials generated by the CbCS management device server (see 5.13.6.8.3). The mechanism for coordination between the CbCS management device server and the CbCS management application client is not defined in this standard.

…

{{At the bottom of figure 15, change the Receive shared keys() operation in the Enforcement Manager to Exchange shared keys() and change the Send shared keys() operation in the CbCS Management Application Client to Exchange shared keys(). This is necessary to correctly reflect the nature of the Diffie-Hellman protocol used to manage the shared keys. Also change Receive/Send security settings() to Receive/Send CbCS parameters. This is necessary to align the figure with the nomenclature used in this proposal.}}

…

5.13.6.8.2 Security Manager class

The Security Manager class for the CbCS technique manages access of secure CDB originators to logical units or volumes (see SSC-3). It uses a decision database to obtain the authorization information required for deciding the type and duration of access granted to secure CDB originator to a given logical unit (see 3.1.71) or volume (see SSC-3). It communicates with secure CDB originators to provide them CbCS credentials ~~(see x.x.x)~~ (see 6.r.2.3), and with enforcement managers (see 3.1.38) to ~~provide them~~ exchange shared keys (see 5.13.6.8.11) ~~(see 3.1.147)~~ and security ~~settings~~ parameters (see 5.13.6.8.15).

…

5.13.6.8.3 CbCS Management Device Server class

5.13.6.8.3.1 CbCS Management Device Server class overview

The CbCS Management Device Server class returns a CbCS capability and a CbCS capability key (i.e., Capability-Key) with each CbCS credential giving the secure CDB originator access to a specific logical unit, and optionally to a volume (see SSC-3).

This standard defines the RECEIVE CREDENTIAL command (see 6.r) that the secure CDB originator may use to request a CbCS capability and a CbCS capability key from a CbCS management device server.

The CbCS management device server shall authenticate the secure CDB originator unless the BASIC CbCS method (see x.x.x) is used ~~(see x.x.x)~~.

…

5.13.6.8.4 CbCS Management Application Client class

The CbCS Management Application Client class ~~sends~~ exchanges shared keys (see 5.13.6.8.11) ~~(see x.x.x)~~ with and sends CbCS parameters (see 5.13.6.8.15) to ~~security settings to~~ the enforcement manager using SECURITY PROTOCOL OUT commands (see 6.31) and SECURITY PROTOCOL IN commands (see 6.30) sent over the SCSI domain's service delivery subsystem.

The CbCS capability keys are computed by the CbCS management device server using shared keys that are shared between the:

    a) Enforcement manager;
    b) CbCS management application client; and
    c) CbCS management device server.

The shared keys are managed by the security manager. ~~This standard defines SCSI commands (see clause 6) the CbCS management application client may use to set and manage the shared keys stored in an enforcement manager.~~

{{The commands are SECURITY PROTOCOL IN and SECURITY PROTOCOL OUT, and they are already mentioned by name in the first paragraph of 5.13.6.8.4.}}

5.13.6.8.5 Secure CDB Originator class

The Secure CDB Originator class requests CbCS capabilities and CbCS capability keys from the CbCS management device server for a specific logical unit (see 3.1.71) or volume (see SSC-3). The secure CDB originator sends the CbCS capability ~~(see x.x.x)~~ and ~~CbCS validation tag~~ integrity check value ~~(see x.x.x)~~ to the logical unit's secure CDB processor as part of a CbCS extended CDB as described in 5.13.6.8.16.

For more information on the Secure CDB Originator class see 5.13.6.2.

5.13.6.8.6 Secure CDB Processor class

The Secure CDB Processor class:

    a)   Receives a CbCS ~~capability~~ (see 6.r.2.3) ~~(see x.x.x)~~ from a secure CDB originator;
    b)   Requests the SCSI command be validated by the enforcement manager; and
    c)   If the Enforcement Manager validates the SCSI command, then the secure CDB processor processes that SCSI command.

The secure CDB processor indicates that CbCS is applied to a logical unit by setting the CBCS bit to one in the Extended INQUIRY Data VPD page (see 7.7.4). If the CbCS bit is set to one, the logical unit shall support the following:

    a)   Extended CDBs (see 4.3.4);
    b)   CbCS extension type (see 5.13.6.8.16) ~~(see x.x.x)~~;
    c)   SECURITY PROTOCOL IN commands (see 6.30) specifying the CbCS security protocol (see 7.6.c) ~~(see x.x.x)~~; and
    d)   SECURITY PROTOCOL OUT commands (see 6.31) specifying the CbCS security protocol (see 7.6.c) ~~(see x.x.x)~~.

For more information on the Secure CDB Processor class see 5.13.6.3.

> ~~Editors Note 2 - ROW: The CBCS bit described above is not currently defined in the Extended INQUIRY Data VPD page. Its definition appears in the yet-to-be-approved 08-145.~~

5.13.6.8.7 Enforcement Manager class

The Enforcement Manager class:

    a)   Receives shared keys (see 5.13.6.8.11) ~~(see 3.1.147)~~ and other ~~security settings (see x.x.x)~~ CbCS parameters (see 5.13.6.8.15) from the CbCS management application client;
    b)   Authenticates the CbCS capability received from a secure CDB processor ~~(see x.x.x)~~ with an integrity check value ~~(see x.x.x) using the CbCS capability received from a secure CDB processor:~~ as described in 5.13.6.8.13.2; ~~and~~
    c)   Validates SCSI commands sent by secure CDB originators~~.~~; and
    d)   If the CAPKEY CbCS method (see 5.13.6.8.8.3) is supported, supplies one security token (see 5.13.6.8.10) for each active I_T_L nexus to the secure CDB processor for deliver to the secure CDB originator that is using that I_T_L nexus.

The enforcement manager may be contained within the secure CDB processor, or within the SCSI target device. If the enforcement manager is contained within the secure CDB processor then, the shared keys and CbCS parameters ~~security settings~~ it uses pertain to ~~the~~ that logical unit (see 3.1.71) ~~(see x.x.x)~~. If the enforcement manager is contained within the SCSI target device then, the shared keys and CbCS parameters ~~security settings~~ it uses pertain to the SCSI target device, and the ~~security protocol~~ SECURITY PROTOCOL well-known logical unit (see 8.5) is used for the commands ~~to set~~ that exchange shared keys and set CbCS parameters ~~security settings~~.

If ~~the~~ a shared key is stored in a well known logical unit then ~~the~~ that key is shared between all logical units within the SCSI target device but shall only used by a logical unit if there has been no shared key assigned to that logical unit (i.e., a shared key assigned to a logical unit always overrides any shared key assigned to a well known logical unit).

For more information on the Enforcement Manager class see 5.13.6.4.

5.13.6.8.8 CbCS methods

{{All of 5.13.6.8.8 is new. Editing markups suspended.}}

5.13.6.8.8.1 Overview

The CbCS methods defined by this standard are summarized in table x1.

Table x1 — CbCS methods

CbCS method	Protection provided by the enforcement manager and secure CDB processor	Level of security provided, including I_T nexus security provided by SCSI transport (e.g., FC-SP)	
		Without I_T nexus security	**With I_T nexus security**
BASIC	Protection against errors provided by verifying that the CbCS capability allows processing, but no validation of the authenticity of the CbCS capability.	No protection against attacks	Same protection as is provided by the SCSI transport on the I_T nexus
CAPKEY	Protection against errors and some attacks by both verifying that the CbCS capability allows processing, and validating the authenticity of the CbCS capability.	CbCS capability authenticity assured, but still subject to network attacks (e.g., replay attacks)	CbCS capability authenticity assured, network attacks (e.g., replay attacks) thwarted by SCSI transport security on the I_T nexus

If a secure CDB processors receives a command for a logical unit that has CbCS enabled, the enforcement manager shall validate the command as described in 5.13.6.8.13.2 before any other field in the CDB is validated, including the operation code.

5.13.6.8.8.2 The BASIC CbCS method

The BASIC CbCS method validates that the CbCS capability authorizes the encapsulated command for each CDB. It provides centrally-managed policy-driven command access control mechanism that enforces authorized access based on capabilities.

The BASIC CbCS method does not validate the authenticity of the CbCS capability.

Preparing CbCS credentials for the BASIC CbCS method does not require the knowledge of CbCS shared keys and may be done by the secure CDB originator without coordination with the CbCS management device server. In the CbCS extension descriptor (see 5.13.6.8.16):

   a)   The CbCS capability descriptor (see 6.r.2.3) CBCS METHOD field is set to BASIC;
   b)   The following CbCS capability descriptor fields are ignored:
      A)   The KEY VERSION field; and
      B)   The INTEGRITY CHECK VALUE ALGORITHM field;
      and
   c)   The INTEGRITY CHECK VALUE field is set to zero.

The BASIC CbCS method controls access between the secure CDB originator and the secure CDB processor without requiring authentication of the secure CDB originator. It is sufficient for the secure CDB processor to request that the enforcement manager verify the CbCS capabilities sent by the secure CDB originator.

5.13.6.8.8.3 The CAPKEY CbCS method

The CAPKEY CbCS method provides centrally-managed policy-driven command access control mechanism that enforces authorized access based on capabilities.

In addition, the CAPKEY CbCS method assures the integrity and authenticity of the CbCS capability transferred with each command.

The CAPKEY CbCS method provides for security of commands delivered to the secure CDB processor. When used in conjunction with a secure service delivery subsystem, it provides additional protection against network attacks (see 5.13.6.8.8.1).

capability

capkey

security token

5.13.6.8.9 CbCS trust assumptions

{{All of 5.13.6.8.9 is new. Editing markups suspended.}}

After the logical unit is trusted (i.e., after a secure CDB originator authenticates that it is communicating with a specific logical unit), the secure CDB originator trusts the secure CDB processor and the enforcement manager to do the following:

a) Deny any access attempt from any application client that is not authorized by the security manager; and
b) Deny access from any application client that does not perform the CbCS protocols and functions defined by this standard.

The CbCS management device server and the CbCS management application client are trusted after:

a) The CbCS management device server is authenticated by the secure CDB originator; and
b) The CbCS management application client is authenticated by the CbCS Enforcement Manager.

The CbCS management device server and the CbCS management application client are trusted to do the following:

a) Securely store long-lived shared keys and capability keys;
b) Grant credentials to secure CDB originators according to access control policies that are outside the scope of this standard; and
c) Perform the defined security functions.

The service delivery subsystem between the secure CDB originator and the secure CDB processor is not trusted. However, the CbCS security model for CbCS methods other than BASIC is defined so that commands generated by the secure CDB originator are processed by the secure CDB processor only after the secure CDB originator interacts with both the CbCS management device server and the secure CDB processor as defined in this standard.

The communications trust requirements shown in table x2 provide a basis for the CbCS trust assumptions.

Table x2 — CbCS communications trust requirement

For connections between	CbCS cryptographic communications trust requirements	Requirement level
secure CDB originator and secure CDB processor	Message integrity [b]	Optional [a]
secure CDB originator and CbCS management application client	Message confidentiality [c] and integrity [b]	Mandatory
CbCS management application client and enforcement manager	Message integrity [b]	Mandatory
CbCS management application client and CbCS management device server	Message confidentiality [c] and integrity [b]	Mandatory

[a] If this requirement is not met, then the conditions that table x1(see 5.13.6.8.8) show for an I_T nexus without security apply.

[b] Message integrity algorithms are those that table 79 (see 5.13.8) describes as integrity checking (i.e., AUTH) algorithms.

[c] Message confidentiality algorithms are those that table 79 (see 5.13.8) describes as encryption algorithms.

5.13.6.8.10 CbCS security tokens

{{All of 5.13.6.8.10 is new. Editing markups suspended.}}

A CbCS security token is a random nonce (see 3.1.107) that is chosen by the enforcement manager (see 5.13.6.8.7) and returned to a secure CDB originator (see 5.13.6.8.5) by the secure CDB processor (see 5.13.6.8.6) in response to a SECURITY PROTOCOL IN command (see 6.30) with the SECURITY PROTOCOL field set to 07h (i.e., CbCS) and the SECURITY PROTOCOL SPECIFIC field set to 3Fh (i.e., the Security Token CbCS page) as described in 7.6.c.3.4.

For the convenience of the CbCS Management Application Client (see 5.13.6.8.4), the security token is also returned in the Current CbCS Parameters CbCS page (see 7.6.c.3.5).

The security token shall be unique to each instance of an I_T_L nexus known to the secure CDB processor and enforcement manager. Security tokens shall be reset and maintained as described in this subclause.

Each security token shall contain at least as many bytes as the largest cipher block size for all the integrity checking algorithms supported by the SCSI target device (see 7.6.c.3.3).

If a hard reset, logical unit reset, or I_T nexus loss is detected by the secure CDB processor, the enforcement manager shall be notified of the event once for each affected I_T_L nexus. In response to such a notification the enforcement manager shall discard the security token, if any, associated with the affected I_T_L nexus.

After a power on, the enforcement manager shall discard all security tokens, if any, that it had been maintaining before the power on.

In response to a request for a security token for a given I_T_L nexus from the secure CDB processor, the enforcement manager shall do one of the following:

a) Return the security token value, if any, that is being maintained for the specified I_T_L nexus to the secure CDB processor; or
b) If no security token is being maintained for the specified I_T_L nexus, then:
    1) A security token random nonce shall be prepared;
    2) The security token shall be returned to the secure CDB processor; and
    3) The security token shall be maintained in association with the specified I_T_L nexus until one of the events described in this subclause cause it to be discarded.

{{Since the enforcement manager is allowed to reside in a SCSI target device (i.e., not in a logical unit), this is one of the rare cases in SPC-4 where I_T nexus is not the same as I_T_L nexus, and the difference matters.}}

5.13.6.8.11 CbCS shared keys

{{All of 5.13.6.8.11 is new. Editing markups suspended.}}

5.13.6.8.11.1 Overview

Cryptographic integrity checking for CbCS capabilities depends on the following hierarchy of shared keys (see 3.1.147) that is specific to the CbCS model:

1) A master key that is composed of:
    A) An authentication key; and
    B) A generation key;
    and
2) Up to 16 working keys whose values are generated from the generation key component of the master key.

Each CbCS shared key set shall accommodate:

a) One master key; and
b) At least two working keys.

Coordinated sets of CbCS shared keys that conform to this hierarchy are maintained by:

a) The CbCS management application client (see 5.13.6.8.4);
b) The CbCS management device server (see 5.13.6.8.3); and
c) The enforcement manager (see 5.13.6.8.7).

The mechanism for coordinating CbCS shared key sets between the CbCS management application client and the CbCS management device server is outside the scope of this standard.

The following security protocols are provided by this standard for coordinating CbCS shared key sets between the CbCS management application client and the enforcement manager:

a) A Diffie-Hellman key exchange protocol for changing all the components of the master key (see 5.13.6.8.11.3); and
b) A mechanism that set a working key based on the generation key component of the current master key (see 5.13.6.8.11.4).

These key management protocols associate a key identifier with each shared key (i.e., master key or working key). The CbCS management application client may use these key identifiers to describe the shared key in some way

(e.g., when the shared key was last refreshed or how the intended use of the shared key). Key identifiers shall not be used to contain shared key values.

Within any SCSI target device that contains an enforcement manager, sets of CbCS shared keys are maintained as follows:

a) A separate set of per-logical unit CbCS shared keys for a logical unit that has CbCS enabled;
b) One target-wide set of CbCS shared keys that are accessible to all logical units that have CbCS enabled; or
c) Both a target-wide set of CbCS shared keys and per-logical unit sets of CbCS shared keys.

The enforcement manager in any logical unit that has CbCS enabled shall have access to at least one set of CbCS shared keys.

If an enforcement manager has access to both a target-wide set of CbCS shared keys and a per-logical unit set of CbCS shared keys, then a working key defined in the per-logical unit set of CbCS shared keys, if any, shall be used instead of the equivalent working key from the target-wide set of CbCS shared keys.

All CbCS shared keys should be retired from active use (i.e., set, changed, or discarded) often enough to thwart key guessing attacks (e.g., working keys should be retired about once per hour). Details of this process are outside the scope of this standard.

The shared keys in a CbCS shared key set and the CbCS pages used to manage them between the CbCS management application client and the enforcement manager are summarized in table x3.

Table x3 — Summary of CbCS shared keys

CbCS shared key	Applicable CbCS page	Data direction	Capability protection	Reference
Master Authentication, and Generation	Current CbCS Parameters [a]	In	Working key	7.6.c.3.5
	Set Master Key – Seed Exchange Set Master Key – Seed Exchange Set Master Key – Change Master Key	Out In Out	Authentication key	7.6.c.5.4 7.6.c.3.6 7.6.c.5.5
Working key	Current CbCS Parameters [a]	In	Working key	7.6.c.3.5
	Set Key	Out	Authentication key	7.6.c.5.3
[a] Only key identifiers are returned, not shared key values.				

5.13.6.8.11.2 Specifying which shared key to change

The logical unit to which a SECURITY PROTOCOL IN command (see 6.30) or a SECURITY PROTOCOL OUT command (see 6.31) that specifies one of the CbCS pages shown in table x3 (see 5.13.6.8.11.1) is addressed determines which CbCS shared key is modified as follows:

a) If the command is addressed to the SECURITY PROTOCOL well known logical unit (see 8.5), then the CbCS the target-wide (see 5.13.6.8.11.1) set of CbCS shared keys is modified; or
b) If the command is addressed to a logical unit that is not a well known logical unit, then the per-logical unit (see 5.13.6.8.11.1) set of CbCS shared keys for the specified logical unit are modified.

5.13.6.8.11.3 Updating a master key

5.13.6.8.11.4 Establishing a working key

5.13.6.8.12 CbCS credentials

{{All of 5.13.6.8.12 is new. Editing markups suspended.}}

5.13.6.8.12.1 Overview

Each CbCS credential authorized access to:

a)   A logical unit (see 3.1.71); or
b)   A specific volume (see SSC-3) mounted in a specific logical unit.

The use of CbCS credentials with the BASIC CbCS method (see 5.13.6.8.8.2) is outside the scope of this standard.

The format of a CbCS credential is described in 6.r.2.2.

The primary components of a CbCS credential are as follows:

a)   A CbCS capability whose format and preparation are described in 5.13.6.8.13; and
b)   The CbCS capability key that is an integrity check value (see 3.1.64) that is computed as follows:
  A)   If the credential is to be sent to a secure CDB originator (see 5.13.6.8.5), the CbCS capability key is computed based on a working key as described in 5.13.6.8.12.2; or
  B)   If the credential is being prepared for use by the CbCS management application client (see 5.13.6.8.4), the CbCS capability key is computed based on a working key or the master key as described in 5.13.6.8.12.3.

Regardless of how it is computed, the CbCS capability key is used as follows:

a)   By the secure CDB originator to prepare CbCS extension descriptors (see 5.13.6.8.16) sent to the secure CDB processor (see 5.13.6.8.6);
b)   By the CbCS management application client to prepare CbCS extension descriptors sent to the enforcement manager (see 5.13.6.8.7); and
c)   By the enforcement manager to validate CbCS extension descriptors received by the secure CDB processor.

5.13.6.8.12.2 CbCS capability key computations for the secure CDB originator

For credentials sent to the secure CDB originator (see 5.13.6.8.5), the CbCS capability key (see 5.13.6.8.12.1) is computed without knowledge of the command for which it is being prepared using the following inputs:

a)   The integrity check value algorithm specified by the INTEGRITY CHECK VALUE ALGORITHM field (see 6.r.2.3) in the CbCS capability descriptor; and
b)   The following inputs to this integrity check value algorithm:
  A)   All the bytes in the CbCS capability descriptor (see 6.r.2.3); and
  B)   The working key specified by the KEY VERSION field (see 6.r.2.3) in the CbCS capability descriptor.

5.13.6.8.12.3 CbCS capability key computations for general use

The computation of the CbCS capability key depends on the command for which the CbCS capability key is being computed as described in this subclause. This computation is more general than the computation described in 5.13.6.8.12.2, but it produces the same results because the secure CDB originator should not be allowed to use the commands that produce the exceptional cases described in this subclause.

The CbCS capability key computations described in this subclause are used:

a) For credentials that are to be used by the CbCS management application client (see 5.13.6.8.4); and
b) By the enforcement manager (see 5.13.6.8.7) when validating a CbCS capability descriptor (see 5.13.6.8.13).

The mechanism by which a CbCS management application client command to be associated with the credential is determined is outside the scope of this standard.

When the enforcement manager is validating a CbCS capability descriptor, the command is determined by inspecting the CDB that is being processed.

Based on the command associated with the credential, the CbCS capability key (see 5.13.6.8.12.1) is computed using the following inputs:

a) The integrity check value algorithm specified by the INTEGRITY CHECK VALUE ALGORITHM field (see 6.r.2.3) in the CbCS capability descriptor; and
b) The following inputs to this integrity check value algorithm:
    A) All the bytes in the CbCS capability descriptor (see 6.r.2.3); and
    B) The following CbCS shared key:
        a) If the command is a SECURITY PROTOCOL IN command (see 6.30) or a SECURITY PROTOCOL OUT command (see 6.30) with the SECURITY PROTOCOL field set to 07h (i.e., CbCS) and the SECURITY PROTOCOL SPECIFIC field set to a value that is greater than 7Fh, then the authentication key component of the master key (see 5.13.6.8.11); or
        b) If the command is not one of those described in step b) B) a), then the working key specified by the KEY VERSION field (see 6.r.2.3) in the CbCS capability descriptor.

5.13.6.8.13 CbCS capability descriptors

{{All of 5.13.6.8.13 is new. Editing markups suspended.}}

5.13.6.8.13.1 Overview

CbCS capability descriptors are components of:

a) CbCS credentials (see 5.13.6.8.12); and
b) CbCS extension descriptors (see 5.13.6.8.16).

The format of a CbCS capability descriptor is described in 6.r.2.3.

CbCS capability descriptors contain:

a) Information about what commands are allowed when the CbCS capability descriptor is associated with a specific CDB via a CbCS extension descriptor;
b) Information that identifies a specific logical unit (see 3.1.71) or a specific volume (see SSC-3) mounted in a specific logical unit to which the CbCS capability is bound;
c) An optional time limit on the validity of the CbCS capability descriptor;

   d) Information that the CbCS management application client (see 5.13.6.8.4) may use to invalidate one or more CbCS capability descriptors before the time limit expires; and
   e) Information that the enforcement manager uses to cryptographically validate the CbCS capability descriptor, if specified, as described in 5.13.6.8.13.

5.13.6.8.13.2 CbCS extension descriptor validation

The enforcement manager (see 5.13.6.8.7) shall validate the CbCS capability descriptor (see 6.r.2.3) included in the CbCS extension descriptor (see 3.1.a). If the validation fails, the enforcement manager shall interact with the secure CDB processor (see 5.13.6.8.6) in a way that causes the command containing the CbCS extension descriptor to be terminated with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

The enforcement manager's validation of a CbCS extension descriptor shall fail if any of the following conditions occur:

   a) A CbCS extension descriptor is not present on a command that table x4 (see 5.13.6.8.14) shows as requiring a CbCS capability;
   b) The command is one that table x4 (see 5.13.6.8.14) shows as never being allowed when CbCS is enabled;
   c) The CBCS METHOD field is set to a value that is less than the value in the minimum CbCS method CbCS parameter (see 5.13.6.8.15);
   d) The CBCS METHOD field contains a value that table x15 defines as reserved (see 6.r.2.3) or a value that the enforcement manager does not support (see 7.6.c.3.3);
   e) If the CBCS METHOD field is set to CAPKEY and the validation described in 5.13.6.8.13.3 fails;
   f) The DESIGNATION TYPE field contains a value that table x14 defines as reserved (see 6.r.2.3);
   g) The DESIGNATION TYPE field value is set to 1h (i.e., logical unit designation descriptor), and the contents of the DESIGNATION DESCRIPTOR field in which a logical unit name (see SAM-4) is indicated does not match the addressed logical unit;
   h) The DESIGNATION TYPE field value is set to 2h (MAM attribute descriptor) and either of the following are true:
      A) The ATTRIBUTE IDENTIFIER field in the DESIGNATION DESCRIPTOR field contains any value other than 0401h (i.e., MEDIUM SERIAL NUMBER); or
      B) The DESIGNATION DESCRIPTOR field contents do not match the MAM attribute of the volume that is accessible via the addressed logical unit;
   i) The CAPABILITY EXPIRATION TIME field contains a non-zero value and the value in the CAPABILITY EXPIRATION TIME field is greater than (i.e., after) the current time in the clock CbCS parameter (see 5.13.6.8.15);
   j) The POLICY ACCESS TAG field contains a non-zero value that does not match the policy access tag CbCS parameter (see 5.13.6.8.15); or
   k) The command in the CDB field of the extended CDB (see 4.3.4) that contains the CbCS extension descriptor is not permitted by the PERMISSIONS BIT MASK field (see 5.13.6.8.14).

5.13.6.8.13.3 CAPKEY CbCS method validation

If the CbCS method is CAPKEY, the enforcement manager's validation of a CbCS capability descriptor shall fail if any of the described in this subclause occur.

Before attempting to cryptographically validate the CbCS capability descriptor, the enforcement manager shall fail the validation if any of the following conditions occur in the CbCS capability descriptor (see 6.r.2.3):

   a) The KEY VERSION field specifies an invalid working key as follows:
      A) The command is not a SECURITY PROTOCOL IN command (see 6.30) or a SECURITY PROTOCOL OUT command (see 6.30) with the SECURITY PROTOCOL field set to 07h (i.e., CbCS) and the SECURITY

PROTOCOL SPECIFIC field set to a value that is greater than 7Fh (i.e., the key version field is ignored for these commands);
  B) The per-logical unit working key (see 5.13.6.8.11), if any, specified by the KEY VERSION field is undefined; and
  C) The target-wide working key (see 5.13.6.8.11), if any, specified by the KEY VERSION field is undefined; and
 b) The INTEGRITY CHECK VALUE ALGORITHM field contains a value that is:
  A) Not one of those that table 71 (see 5.13.8) lists as being an integrity checking (i.e., AUTH) algorithm;
  B) Is AUTH_COMBINED; or
  C) Is a value that the enforcement manager does not support (see 7.6.c.3.3).

If no errors are found in the CbCS capability descriptor, the enforcement manager shall:

  1) Compute the CbCS capability key for the CbCS capability descriptor as described in 5.13.6.8.12.3; and
  2) Compute the expected contents of CbCS extension descriptor INTEGRITY CHECK VALUE field (see 5.13.6.8.16), using the following inputs:
  A) The integrity check value algorithm specified by the INTEGRITY CHECK VALUE ALGORITHM field (see 6.r.2.3) in the CbCS capability descriptor;
  B) The following inputs to this integrity check value algorithm:
  a) All the bytes in the security token (see 5.13.6.8.10) for the I_T_L nexus on which the command was received;
  b) The CbCS capability key computed in step 1).

The enforcement manager shall fail the validation if the contents of CbCS extension descriptor INTEGRITY CHECK VALUE field do not match the computed expected contents of CbCS extension descriptor INTEGRITY CHECK VALUE field.

5.13.6.8.14 Association between commands and permission bits

{{All of 5.13.6.8.14 is new. Editing markups suspended.}}

The PERMISSIONS BIT MASK field in the CbCS capability (see 6.r.2.3) specifies which commands are allowed by the CbCS capability. When processing commands with the CbCS extension, the enforcement manager shall verify that the bits applicable to the encapsulated SCSI command are all set to one in the PERMISSIONS BIT MASK field before processing the command. The associations between commands and permission bits are specified in table x4 for commands defined in this standard.

Table x4 — Associations between commands and permissions (part 1 of 3)

Command	PERMISSIONS BIT MASK bits [a]						
	DATA READ	DATA WRITE	PARM READ	PARM WRITE	SEC MGMT	RESRV	MGMT
ACCESS CONTROL IN	never allow [b]						
ACCESS CONTROL OUT	never allow [b]						
CHANGE ALIASES	always allow [c]						
EXTENDED COPY	never allow [b]						
INQUIRY	always allow [c]						
LOG SELECT				1			
LOG SENSE			1				
MANAGEMENT PROTOCOL IN							1
MANAGEMENT PROTOCOL OUT							1
MODE SELECT(6)				1			
MODE SELECT(10)				1			
MODE SENSE(6)			1				
MODE SENSE(10)			1				
PERSISTENT RESERVE IN			1				
PERSISTENT RESERVE OUT						1	

[a] The command in the CDB field of the extended CDB (see 4.3.4) that contains the CbCS extension descriptor shall be allowed only when all of the bits marked with a 1 in the row for that command are set in the PERMISSIONS BIT MASK field of the CbCS capability in the CbCS extension descriptor. The permissions bits represented by the empty cells in a row are ignored.

[b] If the CBCS bit is set to one in the Extended INQUIRY Data VPD page (see 7.7.4), this command shall never be allowed.

[c] This command shall always be allowed regardless of whether the CbCS extension descriptor is present and if the CbCS extension descriptor is present regardless of the value in the PERMISSIONS BIT MASK field.

[d] This row applies to all identified SECURITY PROTOCOL IN commands except those with the SECURITY PROTOCOL field set to 07h (i.e., CbCS) and the SECURITY PROTOCOL SPECIFIC field set to a value that is less than 40h. SECURITY PROTOCOL IN commands with the SECURITY PROTOCOL field set to 07h (i.e., CbCS) and the SECURITY PROTOCOL SPECIFIC field set to a value that is less than 40h shall always be allowed regardless of whether the CbCS extension descriptor is present and if the CbCS extension descriptor is present regardless of the value in the PERMISSIONS BIT MASK field. {{This is necessary to allow a secure CDB originator to determine the security token for CAPKEY computations need for a capkey which is required for such computations.}}

Table x4 — Associations between commands and permissions (part 2 of 3)

Command	PERMISSIONS BIT MASK bits [a]						
	DATA READ	DATA WRITE	PARM READ	PARM WRITE	SEC MGMT	RESRV	MGMT
READ ATTRIBUTE			1				
READ BUFFER					1		
READ MEDIA SERIAL NUMBER			1				
RECEIVE COPY RESULTS	never allow [b]						
RECEIVE CREDENTIAL	always allow [c]						
RECEIVE DIAGNOSTIC RESULTS			1				
REPORT ALIASES			1				
REPORT IDENTIFYING INFORMATION			1				
REPORT LUNS	always allow [c]						
REPORT PRIORITY			1				
REPORT SUPPORTED OPERATION CODES	always allow [c]						
REPORT SUPPORTED TASK MANAGEMENT FUNCTIONS	always allow [c]						
REPORT TARGET PORT GROUPS	always allow [c]						
REPORT TIMESTAMP			1				
REQUEST SENSE			1				
SECURITY PROTOCOL IN with the SECURITY PROTOCOL field set to 00h	always allow [c]						
SECURITY PROTOCOL IN with the SECURITY PROTOCOL field not set to 00h [d]					1		
SECURITY PROTOCOL OUT					1		
SEND DIAGNOSTIC				1			
SET IDENTIFYING INFORMATION				1			

[a] The command in the CDB field of the extended CDB (see 4.3.4) that contains the CbCS extension descriptor shall be allowed only when all of the bits marked with a 1 in the row for that command are set in the PERMISSIONS BIT MASK field of the CbCS capability in the CbCS extension descriptor. The permissions bits represented by the empty cells in a row are ignored.

[b] If the CBCS bit is set to one in the Extended INQUIRY Data VPD page (see 7.7.4), this command shall never be allowed.

[c] This command shall always be allowed regardless of whether the CbCS extension descriptor is present and if the CbCS extension descriptor is present regardless of the value in the PERMISSIONS BIT MASK field.

[d] This row applies to all identified SECURITY PROTOCOL IN commands except those with the SECURITY PROTOCOL field set to 07h (i.e., CbCS) and the SECURITY PROTOCOL SPECIFIC field set to a value that is less than 40h. SECURITY PROTOCOL IN commands with the SECURITY PROTOCOL field set to 07h (i.e., CbCS) and the SECURITY PROTOCOL SPECIFIC field set to a value that is less than 40h shall always be allowed regardless of whether the CbCS extension descriptor is present and if the CbCS extension descriptor is present regardless of the value in the PERMISSIONS BIT MASK field. {{This is necessary to allow a secure CDB originator to determine the security token for CAPKEY computations need for a capkey which is required for such computations.}}

Table x4 — Associations between commands and permissions (part 3 of 3)

Command	PERMISSIONS BIT MASK bits [a]						
	DATA READ	DATA WRITE	PARM READ	PARM WRITE	SEC MGMT	RESRV	MGMT
SET PRIORITY				1			
SET TARGET PORT GROUPS				1			
SET TIMESTAMP				1	1		
TEST UNIT READY	always allow [c]						
WRITE ATTRIBUTE				1			
WRITE BUFFER					1		

[a] The command in the CDB field of the extended CDB (see 4.3.4) that contains the CbCS extension descriptor shall be allowed only when all of the bits marked with a 1 in the row for that command are set in the PERMISSIONS BIT MASK field of the CbCS capability in the CbCS extension descriptor. The permissions bits represented by the empty cells in a row are ignored.

[b] If the CBCS bit is set to one in the Extended INQUIRY Data VPD page (see 7.7.4), this command shall never be allowed.

[c] This command shall always be allowed regardless of whether the CbCS extension descriptor is present and if the CbCS extension descriptor is present regardless of the value in the PERMISSIONS BIT MASK field.

[d] This row applies to all identified SECURITY PROTOCOL IN commands except those with the SECURITY PROTOCOL field set to 07h (i.e., CbCS) and the SECURITY PROTOCOL SPECIFIC field set to a value that is less than 40h. SECURITY PROTOCOL IN commands with the SECURITY PROTOCOL field set to 07h (i.e., CbCS) and the SECURITY PROTOCOL SPECIFIC field set to a value that is less than 40h shall always be allowed regardless of whether the CbCS extension descriptor is present and if the CbCS extension descriptor is present regardless of the value in the PERMISSIONS BIT MASK field. {{This is necessary to allow a secure CDB originator to determine the security token for CAPKEY computations need for a capkey which is required for such computations.}}

Command standards (see 3.1.18) may describe the associations between commands and permission bits for the commands that they define. The processing requirements for those associations are the same as those described in this standard.

{{The paragraph immediately above is much less wordy than 07-454r5, but it should be saying the same thing. Reviewers should verify this.}}

5.13.6.8.15 CbCS parameters

{{All of 5.13.6.8.15 is new. Editing markups suspended.}}

5.13.6.8.15.1 Overview

CbCS parameters:

   a) Provide the CbCS Management Application Client class (see 5.13.6.8.4) with a means to control the operation of the Enforcement Manager class(see 5.13.6.8.7); and
   b) Provide a Diffie-Hellman protocol by which shared keys may be established between the CbCS Management Application Client class and the Enforcement Manager class;
   c) Allow the Secure CDB Processor class(see 5.13.6.8.6) to receive security tokens and other CbCS information from the Enforcement Manager class; and

  d) Allow any application client to receive basic operational CbCS information from the Enforcement Manager
      class.

CbCS parameters that are not changeable indicate which CbCS features and algorithms are supported. An application client may retrieve the unchangeable CbCS parameters by using the SECURITY PROTOCOL IN command to return the Unchangeable CbCS Parameters CbCS page (see 7.6.c.3.3).

The CbCS parameters with values that change in response to various conditions are summarized in table x5.

Table x5 — Summary of changeable CbCS parameters

Parameter	Support	Applicable CbCS page	Data direction	Capability protection	Reference
CbCS parameters that are updated automatically based on I_T nexus					
Security token	Mandatory [a]	Security Token	In	None	7.6.c.3.4
		Current CbCS Parameters	In	Working key	7.6.c.3.5
CbCS parameters that provide initial values for dynamically created logical units					
Initial minimum CbCS method		...			
Initial policy access tag					
CbCS parameters that affect the CbCS enforcement manager processing					
Minimum CbCS method	Mandatory	Current CbCS Parameters	In	Working key	7.6.c.3.5
		Set Minimum CbCS Method	Out	Working key	7.6.c.5.2
Policy Access Tag	Mandatory	Current CbCS Parameters	In	Working key	7.6.c.3.5
		Set Policy Access Tag	Out	Working key	7.6.c.5.1
Clock	Mandatory	Current CbCS Parameters	In	Working key	7.6.c.3.5
Shared keys and shared key identifiers	Mandatory	see 5.13.6.8.11			
[a] Mandatory only if the CAPKEY CbCS method (see 5.13.6.8.8.3) is supported.					

5.13.6.8.16 CbCS extension descriptor format

{{All of 5.13.6.8.16 is new. Editing markups suspended.}}

The CbCS extension descriptor (see table x6) allows the capability-based command security technique (see 5.13.6.8) to used with a SCSI command via the parameters specified in this subclause. Support for the CbCS extension descriptor is mandatory if CBCS bit to one in the Extended INQUIRY Data VPD page (see 7.7.4). When an extended CDB (see 4.3.4) includes a CbCS extension descriptor the CDB field may contain any CDB defined in this standard or any SCSI command standard (see 3.1.18).

Table x6 — CbCS extension descriptor format

Bit Byte	7	6	5	4	3	2	1	0
0	EXTENSION TYPE (40h)							
1	Reserved							
3								
4	CbCS capability descriptor (see 6.r.2.3)							
77								
78	(MSB)			INTEGRITY CHECK VALUE				
141								(LSB)

The EXTENSION TYPE field contains 40h (i.e., CbCS extension descriptor).

The CbCS capability descriptor is defined in 6.r.2.3.

The INTEGRITY CHECK VALUE field contains an integrity check value (see 3.1.65).

The CbCS capability descriptor and the INTEGRITY CHECK VALUE field should be prepared by the secure CDB originator as described in 2.4.3.

The enforcement manager shall validate the CbCS capability descriptor and the INTEGRITY CHECK VALUE field as described in 5.13.6.8.13.2.

6.r RECEIVE CREDENTIAL command

{{All of 6.r is new. Editing markups suspended.}}

{{Numerous changes shown in 08-128r3 are replicated in 6.r without identifying annotations.}}

6.r.1 RECEIVE CREDENTIAL command description

6.r.1.1 Overview

The RECEIVE CREDENTIAL command (see table x7) allows a secure CDB originator (see 5.13.6.2) to receive a credential from a security manager device server (e.g., a CbCS management device server (see 5.13.6.8.3)) for use in a CDB (e.g., use in the CbCS extension descriptor (see 3.1.a)).

Table x7 — RECEIVE CREDENTIAL command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (7Fh)							
1	CONTROL							
2 6	Reserved							
7	ADDITIONAL CDB LENGTH (n-7)							
8	(MSB)			SERVICE ACTION (1800h)				
9								(LSB)
10	(MSB)			ALLOCATION LENGTH				
11								(LSB)
12 15	Restricted (see RFC 4306)							
16	(MSB)			AC_SAI				
19								(LSB)
20 23	Restricted (see RFC 4306)							
24	(MSB)			DS_SAI				
27								(LSB)
28	(MSB)			CREDENTIAL REQUEST TYPE				
29								(LSB)
30 n	CREDENTIAL REQUEST DESCRIPTOR							

The ALLOCATION LENGTH field is defined in 4.3.5.6.

The AC_SAI field contains the value of the AC_SAI SA parameter (see 5.13.2.2) for the SA to be used to encrypt the parameter data as described in 6.r.2.1.

The DS_SAI field contains the value of the DS_SAI SA parameter (see 5.13.2.2) for the SA to be used to encrypt the parameter data as described in 6.r.2.1.

If the device server is not maintaining an SA with an AC_SAI SA parameter that matches the AC_SAI field contents and a DS_SAI SA parameter that matches the DS_SAI field contents, then 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.

If the device server is maintaining the SA specified by the AC_SAI field and the DS_SAI field, then the SA shall be verified for use by this RECEIVE CREDENTIAL command as follows:

   a) The USAGE_TYPE SA parameter (see 5.13.2.2) shall be verified to be equal to 82h (i.e., CbCS authentication and credential encryption; and
   b) The USAGE_DATA SA parameter (see 5.13.2.2) shall be verified not to contain an ALGORITHM IDENTIFIER field (see 7.6.3.6) that is set to ENCR_NULL based on the contents the IKEv2-SCSI SAUT Cryptographic Algorithm payload (see 7.6.3.15.13) for the ENCR algorithm type (see 7.6.3.6.2) during creation of the SA (see 5.13.2.3).

If any of these SA verifications fails, 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.

{{The current RECEIVE CREDENTIAL command structure has no mechanism for verifying that the source of the RECEIVE CREDENTIAL command has any knowledge of the shared keys in the specified SA. If source does not know the shared keys, it will not be able to decrypt the parameter data. However, various known plaintext attacks are possible when knowledge of the shared keys is not required to obtain useful information from a RECEIVE CREDENTIAL command. Addressing these attacks will be handled in a separate proposal.}}

The CREDENTIAL REQUEST TYPE field (see table x8) specifies type of credential being requested and the format of the CREDENTIAL REQUEST DESCRIPTOR field.

Table x8 — CREDENTIAL REQUEST TYPE field

Code	Description	Reference
0001h	CbCS logical unit	6.r.1.2
0002h	CbCS logical unit and volume	6.r.1.3
all other codes	Reserved	

{{A row in this table might define a range of codes as restricted to OSD, but first the interests of the SNIA OSD TWG must be assessed.}}

If return of the requested credential is not permitted, the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to ACCESS DENIED - NO ACCESS RIGHTS.

6.r.1.2 CbCS logical unit credential request descriptor

If the credential request type field is set to 0001h (i.e., CbCS logical unit), then the format of the CREDENTIAL REQUEST DESCRIPTOR field is as shown in table x9.

Table x9 — CbCS logical unit credential request descriptor format

Bit Byte	7	6	5	4	3	2	1	0
0				DESIGNATION DESCRIPTOR				
19								

The format of the DESIGNATION DESCRIPTOR field is defined in table 422 (see 7.7.3.1). 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 if any of the fields in the DESIGNATION DESCRIPTOR field are set as follows:

   a) The DESIGNATOR TYPE field contains any value other than 3h (i.e., NAA);
   b) The ASSOCIATION field contains any value other than 00b (i.e., logical unit) or 10b (i.e., SCSI target device); or
   c) The DESIGNATOR LENGTH field is set to a value that is larger than 16.

6.r.1.3 CbCS logical unit and volume credential request descriptor

If the credential request type field is set to 0002h (i.e., CbCS logical unit and volume), then the format of the CREDENTIAL REQUEST DESCRIPTOR field is as shown in table x10.

Table x10 — CbCS logical unit and volume credential request descriptor format

Bit Byte	7	6	5	4	3	2	1	0
0				DESIGNATION DESCRIPTOR				
19								
20				MAM ATTRIBUTE				
56								

The format of the DESIGNATION DESCRIPTOR field is defined in table 422 (see 7.7.3.1). 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 if any of the fields in the DESIGNATION DESCRIPTOR field are set as follows:

   a) The DESIGNATOR TYPE field contains any value other than 3h (i.e., NAA);
   b) The ASSOCIATION field contains any value other than 00b (i.e., logical unit) or 10b (i.e., SCSI target device); or
   c) The DESIGNATOR LENGTH field is set to a value that is larger than 16.

The format of the MAM ATTRIBUTE field is defined in table 310 (see 7.3.1). If the ATTRIBUTE IDENTIFIER field in the MAM ATTRIBUTE field contains any value other than 0401h (i.e., MEDIUM SERIAL NUMBER), 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.

6.r.2 RECEIVE CREDENTIAL parameter data

6.r.2.1 RECEIVE CREDENTIAL parameter data encryption

The RECEIVE CREDENTIAL parameter data shall be one of the ESP-SCSI data-in buffer descriptors shown in table 74 (see 5.13.7.5.1). The SA specified by the AC_SAI field and the DS_SAI field in the CDB shall be used to construct the ESP-SCSI data-in buffer descriptor as described in 5.13.7.5.

Before processing the parameter data, the application client should validate and decrypt the ESP-SCSI data-in buffer descriptor as described in 5.13.7.5. If any errors are detected by the validation and decryption processing, the parameter data should be ignored.

6.r.2.2 RECEIVE CREDENTIAL decrypted parameter data

Before encryption and after decryption, the UNENCRYPTED BYTES field (see 5.13.7.3) that are used to compute the ENCRYPTED OR AUTHENTICATED DATA field (see 5.13.7.5) contents shall contain a CbCS credential (see table x11).

Table x11 — CbCS credential format

Bit Byte	7	6	5	4	3	2	1	0
0	Reserved				CREDENTIAL FORMAT (1h)			
1	Reserved							
2	(MSB)			CREDENTIAL LENGTH (n-3)				
3								(LSB)
4	(MSB)			CAPABILITY LENGTH (k-5)				
5								(LSB)
6				CbCS capability descriptor (see 6.r.2.3)				
k								
k+1	(MSB)			CAPABILITY KEY LENGTH (n-k-4))				
k+4								(LSB)
k+5				CAPABILITY KEY				
n								

{{The CR PRSNT bit defined in 07-454r5 has been removed in this proposal because it is redundant with the length information in the ESP-SCSI descriptor.}}

The CREDENTIAL FORMAT field (see table x12) indicates the format of the credential.

Table x12 — Credential format values

Value	Description
0h	Reserved
1h	The format defined by this standard
2h - Fh	Reserved

{{Because the capability format is not in the capability descriptor, the information it contains is not available in the CbCS extension descriptor. This requires the enforcement manager to use the XCDB extension descriptor EXTENSION TYPE field as the indicator of capability format.}}

The CREDENTIAL LENGTH field indicates the number of bytes that follow in the credential including the capability length, the CbCS capability descriptor, the capability key length, and the capability key.

The CAPABILITY LENGTH field indicates the number of bytes that follow in the capability.

The contents of the CbCS capability descriptor are defined in 6.r.2.3.

The CAPABILITY KEY LENGTH field indicates the number of bytes that follow in the capability key.

The CAPABILITY KEY field contains an integrity check value (see 3.1.64) that is computed and used as described in 5.13.6.8.12.

6.r.2.3 CbCS capability descriptor

6.r.2.3.1 Overview

A CbCS capability descriptor (see table x13) specifies the commands that are allowed by the CbCS extension descriptor in which it appears.

Table x13 — CbCS capability descriptor format

Bit Byte	7	6	5	4	3	2	1	0
0	DESIGNATION TYPE				KEY VERSION			
1	CBCS METHOD							
2	(MSB)			INTEGRITY CHECK VALUE ALGORITHM				
5								(LSB)
6	(MSB)			CAPABILITY EXPIRATION TIME				
11								(LSB)
12				PERMISSIONS BIT MASK				
15								
16	(MSB)			POLICY ACCESS TAG				
19								(LSB)
20				DESIGNATION DESCRIPTOR				
57								
58				DISCRIMINATOR				
73								

{{The designation descriptor field is one byte too short. It is 36 bytes and it needs to be 37 to accommodate a MAM attribute.}}

The DESIGNATION TYPE field (see table x14) specifies the format of the Designation descriptor.

Table x14 — DESIGNATION TYPE field

Code	Description	DESIGNATION DESCRIPTOR field format reference
0h	Reserved	
1h	Logical unit designation descriptor	6.r.2.3.2
2h	MAM attribute designation descriptor	6.r.2.3.3
3h - Fh	Reserved	

The KEY VERSION field specifies which working key (see 5.13.6.8.11), is being used to compute the following values for this CbCS capability:

    a) The capability key (see 5.13.6.8.12); and
    b) The CbCS extension integrity check value (see 5.13.6.8.13).

The CBCS METHOD field (see table x14) specifies the CbCS method used by this CbCS capability.

Table x15 — CBCS METHOD field

Code	CbCS method	Reference
00h	BASIC	5.13.6.8.8.2
01h	CAPKEY	5.13.6.8.8.3
02h - EFh	Reserved	
F0h - FEh	Vendor specific	
FFh	Reserved	

The INTEGRITY CHECK VALUE ALGORITHM field specifies the algorithm used to compute the capability key and other integrity check values for this CbCS capability. The value in the INTEGRITY CHECK VALUE ALGORITHM field is selected from the codes that the Capabilities CbCS page (see 7.6.c.3.3) lists as supported integrity check value algorithms.

The CAPABILITY EXPIRATION TIME field specifies expiration time of this CbCS capability as the number of milliseconds that have elapsed since midnight, 1 January 1970 UT. If the CAPABILITY EXPIRATION TIME field is set to zero, this CbCS capability does not have an expiration time.

If the CAPABILITY EXPIRATION TIME field is not set to zero, then:

a) The clock maintained by the CbCS management device server (see 5.13.6.8.3) should be synchronized with the clock maintained by the enforcement manager (see 5.13.6.8.7). The method for synchronizing the clocks is outside the scope of this standard, however, the protocol should be implemented in a secure manner (e.g., it should not be possible for an adversary to set the clock in the SCSI device or in the secure CDB processor backwards to enable the reuse of expired CbCS credentials). The value in the enforcement manager's clock is available in the Current CbCS Parameters CbCS page (see 7.6.c.3.5) to assist in this synchronization;
b) The CbCS management device server should set the CAPABILITY EXPIRATION TIME field to a value that is at least an order of magnitude larger than the allowed deviation between the clocks.

The PERMISSIONS BIT MASK field (see table x16) specifies the permissions allowed by this CbCS capability. More than one permissions bit may be set. The relationship between commands and bits in the PERMISSIONS BIT MASK field is defined in for the commands defined by this standard and in the command standard (see 3.1.18) that defines commands for a specific device type.

Table x16 — PERMISSIONS BIT MASK field format

Bit Byte	7	6	5	4	3	2	1	0
0	DATA READ	DATA WRITE	PARM READ	PARM WRITE	SEC MGMT	RESRV	MGMT	PHY ACC
1				Reserved				
2								
3				Restricted (see applicable command standard)				

A DATA READ bit set to zero indicates a command has no read permission for user data and protection information. A DATA READ bit set to one indicates a command has permission to read user data and protection information.

A DATA WRITE bit set to zero indicates a command has no write permission for user data and protection information. A DATA WRITE bit set to one indicates a command has permission to write user data and protection information.

A parameter data read (PARM READ) bit set to zero indicates a command has no parameter data read permission. A PARM READ bit set to one indicates a command has permission to read parameter data.

A parameter data write (PARM WRITE) bit set to zero indicates a command has no parameter data write permission. A PARM WRITE bit set to one indicates a command has permission to write parameter data.

A security management (SEC MGMT) bit set to zero indicates a command has no security management permission. A SEC MGMT bit set to one indicates a command has security management permission.

A reservation (RESRV) bit set to zero indicates a command has no persistent reservation permission. A RESRV bit set to one indicates a has permission to make or modify persistent reservations.

A management (MGMT) bit set to zero indicates a command has no storage management permission. A MGMT bit set to one indicates a command has storage management permission. Storage management is outside the scope of this standard.

A physical access (PHY ACC) bit set to zero indicates a command has no permission to affect physical access to the logical unit or volume. A PHY ACC bit set to one indicates a command has permission to affect physical access to the logical unit or volume (see SSC-3).

If the POLICY ACCESS TAG field contains a value other than zero, the policy access tag attribute of the logical unit (see 5.13.6.8.15) is compared to the POLICY ACCESS TAG field contents as part of validating the CbCS capability (see 5.13.6.8.13.2). If the POLICY ACCESS TAG field contains zero, then no comparison is made.

The DESIGNATION DESCRIPTOR field is used during the validation of the CbCS capability (see 5.13.6.8.13.2) to ensure that the command is being addressed to the correct logical unit or volume (see SSC-3). The format of the DESIGNATION DESCRIPTOR field is defined by the value in the DESIGNATION TYPE field as described in table x14. The DESIGNATION DESCRIPTOR field is limited to 36 bytes in length.

If the CREDENTIAL REQUEST TYPE field in a RECEIVE CREDENTIAL command is set to 0001h (i.e., CbCS logical unit), then the DESIGNATION DESCRIPTOR field shall contain a logical unit designation descriptor that matches the DESIGNATION DESCRIPTOR field (see 6.r.1.2) in the CREDENTIAL REQUEST DESCRIPTOR field in the CDB. If the CREDENTIAL REQUEST TYPE field in a RECEIVE CREDENTIAL command is set to 0002h (i.e., CbCS logical unit and volume), then the DESIGNATION DESCRIPTOR field shall contain a MAM attribute designation descriptor that matches the MAM ATTRIBUTE field (see 6.r.1.3) in the CREDENTIAL REQUEST DESCRIPTOR field in the CDB.

The DISCRIMINATOR field provides uniqueness to the CbCS capability descriptor and may be used to limit the delegation or prevent leakage of the CbCS capability to other application clients. The CbCS management device server (see 5.13.6.8.3) shall not return the same CbCS capability descriptor to two secure CDB originators.

The enforcement manager (see 5.13.6.8.7) shall validate each CbCS capability descriptor it receives as described in 5.13.6.8.13.2.

6.r.2.3.2 Logical unit designation descriptor format

If the DESIGNATION TYPE field is set to 0001h (i.e., logical unit designation descriptor), then the format of the DESIG-NATION DESCRIPTOR field is as shown in table x17.

Table x17 — Logical unit designation descriptor format

Bit Byte	7	6	5	4	3	2	1	0
0				DESIGNATION DESCRIPTOR				
19								
20				Reserved				
37								

The format of the DESIGNATION DESCRIPTOR field is defined in table 422 (see 7.7.3.1) with the following additional requirements:

   a) The DESIGNATOR TYPE field shall contain 3h (i.e., NAA);
   b) The ASSOCIATION field shall 00b (i.e., logical unit); and
   c) The DESIGNATOR LENGTH field shall set to a value that is smaller than 17.

6.r.2.3.3 Volume designation descriptors

If the DESIGNATION TYPE field is set to 0002h (i.e., volume unit designation descriptor), then the format of the DESIG-NATION DESCRIPTOR field is as shown in table x18.

Table x18 — Volume designation descriptor format

Bit Byte	7	6	5	4	3	2	1	0
0				MAM ATTRIBUTE				
36								
37				Reserved				

The format of the MAM ATTRIBUTE field is defined in table 310 (see 7.3.1) with the following additional requirements:

   a) The MAM ATTRIBUTE field shall contain 0401h (i.e., MEDIUM SERIAL NUMBER); and
   b) The attribute length field shall contain 0020h.

…

7.6 Security protocol parameters

…

7.6.c CbCS security protocol

{{All of 7.6.c is new. Editing markups suspended.}}

{{Changes related to the INC_512 bit are shown in 08-141r0 are replicated in 7.6.c without identifying annotations.}}

7.6.c.1 Overview

If the SECURITY PROTOCOL field in a SECURITY PROTOCOL IN command (see 6.30) is set to 07h, then the command specifies one of the CbCS pages (see 7.6.c.2) to be returned by the device sever. The information returned by a CbCS SECURITY PROTOCOL IN command indicates the CbCS operating parameters of:

a) The logical unit to which the CbCS SECURITY PROTOCOL IN command is addressed; or
b) The SCSI target device that contains the well-known logical unit to which the CbCS SECURITY PROTOCOL IN command is addressed.

If the SECURITY PROTOCOL field in a SECURITY PROTOCOL OUT command (see 6.31) is set to 07h, then the command specifies one of the CbCS pages (see 7.6.c.4) to be sent to the device sever. The instructions sent in a CbCS SECURITY PROTOCOL OUT command specify the CbCS operating parameters of:

a) The logical unit to which the CbCS SECURITY PROTOCOL OUT command is addressed; or
b) The SCSI target device that contains the well-known logical unit to which the CbCS SECURITY PROTOCOL OUT command is addressed.

7.6.c.2 CbCS SECURITY PROTOCOL IN CDB description

The CbCS SECURITY PROTOCOL IN CDB has the format defined in 6.30 with the additional requirements described in this subclause.

When the SECURITY PROTOCOL field is set to CbCS (i.e., 07h) in a SECURITY PROTOCOL IN command, the SECURITY PROTOCOL SPECIFIC field (see table x19) specifies the CbCS page to be returned in the parameter data (see 7.6.c.3). If the CBCS bit is set to one in the Extended INQUIRY Data VPD page (see 7.7.4), the CbCS SECURITY PROTOCOL IN command support requirements are shown in table x19.

Table x19 — SECURITY PROTOCOL SPECIFIC field for the CbCS SECURITY PROTOCOL IN command

Code	CbCS page returned	Support	Reference
0000h	Supported CbCS SECURITY PROTOCOL IN Pages	Mandatory	7.6.c.3.1
0001h	Supported CbCS SECURITY PROTOCOL OUT Pages	Mandatory	7.6.c.3.2
0002h	Unchangeable CbCS Parameters	Mandatory	7.6.c.3.3
0003h – 003Eh	Reserved		
3Fh	Security Token	Mandatory	7.6.c.3.4
0040h	Current CbCS Parameters	Mandatory	7.6.c.3.5
0041h – CFFFh	Reserved		
D000h	Set Master Key – Seed Exchange	Mandatory	7.6.c.3.6
D001h – FFFFh	Reserved		

{{Code assignments in table x19 have been modified to align them with code assignments in table x27.}}

If a CbCS SECURITY PROTOCOL IN command is received with the INC_512 bit set to one, 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.

7.6.c.3 CbCS SECURITY PROTOCOL IN parameter data

7.6.c.3.1 Supported CbCS SECURITY PROTOCOL IN Pages CbCS page

The Supported CbCS SECURITY PROTOCOL IN Pages CbCS page (see table x20) lists the CbCS pages that are supported for the (i.e., the values that are allowed in the SECURITY PROTOCOL SPECIFIC field in a) SECURITY PROTOCOL IN command (see 6.30).

Table x20 — Supported CbCS SECURITY PROTOCOL IN Pages CbCS page format

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)			PAGE CODE (0000h)				
1								(LSB)
2	(MSB)			PAGE LENGTH (n-3)				
3								(LSB)
	Supported CbCS SECURITY PROTOCOL IN page list							
4	(MSB)			SUPPORTED CBCS SECURITY PROTOCOL IN PAGE [first]				
5				(0000h)				(LSB)
				. . .				
n-1	(MSB)			SUPPORTED CBCS SECURITY PROTOCOL IN PAGE [last]				
n								(LSB)

The PAGE CODE field shall be set to 0000h to indicate that the Supported CbCS SECURITY PROTOCOL IN Pages CbCS page is being returned.

The page length field indicates the number of bytes that follow in the Supported CbCS SECURITY PROTOCOL IN Pages CbCS page.

Each SUPPORTED CBCS SECURITY PROTOCOL IN PAGE field indicates the page code (see table x19 in 7.6.c.2) of one CbCS page that is supported by the SECURITY PROTOCOL IN command when the SECURITY PROTOCOL field (see 6.30) is set to 07h (i.e., CbCS). The values in the SUPPORTED CBCS SECURITY PROTOCOL IN PAGE fields shall be returned in ascending order beginning with 0000h (i.e., this CbCS page).

7.6.c.3.2 Supported CbCS SECURITY PROTOCOL OUT pages CbCS page

The Supported CbCS SECURITY PROTOCOL OUT Pages CbCS page (see table x21) lists the CbCS pages that are supported for the (i.e., the values that are allowed in the SECURITY PROTOCOL SPECIFIC field in a) SECURITY PROTOCOL OUT command (see 6.31).

Table x21 — Supported CbCS SECURITY PROTOCOL OUT Pages CbCS page format

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)							
1				PAGE CODE (0001h)				(LSB)
2	(MSB)							
3				PAGE LENGTH (n-3)				(LSB)
Supported CbCS SECURITY PROTOCOL OUT page list								
4	(MSB)			SUPPORTED CBCS SECURITY PROTOCOL OUT PAGE				
5				[first]				(LSB)
				. . .				
n-1	(MSB)			SUPPORTED CBCS SECURITY PROTOCOL OUT PAGE				
n				[last]				(LSB)

The PAGE CODE field shall be set to 0001h to indicate that the Supported CbCS SECURITY PROTOCOL OUT Pages CbCS page is being returned.

The page length field indicates the number of bytes that follow in the Supported CbCS SECURITY PROTOCOL OUT Pages CbCS page.

Each SUPPORTED CBCS SECURITY PROTOCOL OUT PAGE field indicates the page code (see table x27 in 7.6.c.4) of one CbCS page that is supported by the SECURITY PROTOCOL OUT command when the SECURITY PROTOCOL field (see 6.31) is set to 07h (i.e., CbCS). The values in the SUPPORTED CBCS SECURITY PROTOCOL IN PAGE fields shall be returned in ascending order.

7.6.c.3.3 Unchangeable CbCS Parameters CbCS page

The Unchangeable CbCS Parameters CbCS page (see table x22) indicates the supported CbCS features and algorithms.

Table x22 — Unchangeable CbCS Parameters CbCS page format (part 1 of 2)

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)							
1				PAGE CODE (0002h)				(LSB)
2	(MSB)							
3				PAGE LENGTH (n-3)				(LSB)
4	KEYS SUPPORT		MIN CBCS METHOD SUP		Reserved			
5	Reserved							
6	(MSB)			SUPPORTED INTEGRITY CHECK VALUE ALGORITHM				
7				LIST LENGTH (c-7)				(LSB)
Supported integrity check value algorithms list								
8	(MSB)			SUPPORTED INTEGRITY CHECK VALUE ALGORITHM				
11				[first]				(LSB)
				. . .				
c-3	(MSB)			SUPPORTED INTEGRITY CHECK VALUE ALGORITHM				
c				[last]				(LSB)
c+1				Reserved				
c+2								
c+3	(MSB)			SUPPORTED D-H ALGORITHM LIST LENGTH (d-c-4)				
c+4								(LSB)
Supported Diffie-Hellman (D-H) algorithms list								
c+5	(MSB)			SUPPORTED D-H ALGORITHM [first]				
c+8								(LSB)
				. . .				
d-3	(MSB)			SUPPORTED D-H ALGORITHM [last]				
d								(LSB)
d+1	(MSB)			SUPPORTED CBCS METHODS LIST LENGTH (n-d-2)				
d+2								(LSB)

Table x22 — Unchangeable CbCS Parameters CbCS page format (part 2 of 2)

Bit Byte	7	6	5	4	3	2	1	0
	Supported CbCS methods list							
d+3	SUPPORTED CBCS METHOD [first]							
	. . .							
n	SUPPORTED CBCS METHOD [last]							

The PAGE CODE field shall be set to 0002h to indicate that the Unchangeable CbCS Parameters CbCS page is being returned.

The page length field indicates the number of bytes that follow in the Unchangeable CbCS Parameters CbCS page.

The KEYS SUPPORT field (see table x23) indicates the type of CbCS master keys and working keys supported.

Table x23 — KEYS SUPPORT field

Code	Description
00b	Reserved
01b	The SCSI target device supports single CbCS master key and a set of CbCS working keys (see ?.?.?) for the SCSI target device, but the logical units in the SCSI target device do not support CbCS master keys or working keys.
10b	The SCSI target device does not support CbCS master keys or working keys for the SCSI target device, but each logical unit in the SCSI target device supports a single CbCS master key and a set of CbCS working keys for that logical unit.
11b	The SCSI target device supports single CbCS master key and a set of CbCS working keys for the SCSI target device, and each logical unit in the SCSI target device supports a single CbCS master key and a set of CbCS working keys for that logical unit. Keys stored in the logical unit take precedence over keys stored in the SCSI target device (see 5.13.6.8.6).

The MIN CBCS METHOD SUP field (see table x24) indicates how the assignment of the minimum allowable CbCS method is supported.

Table x24 — MIN CBCS METHOD SUP field

Code	Description
00b	Reserved
01b	A single minimum allowed CbCS method (see ?.?.?) is assigned to all logical units in the SCSI target device, and a SECURITY PROTOCOL well known logical unit (see 8.5) is implemented to manage it.
10b	Each logical unit in the SCSI target device is assigned its own minimum allowed CbCS method.
11b	Reserved

The SUPPORTED INTEGRITY CHECK VALUE ALGORITHM LIST LENGTH field indicates the number of bytes that follow in the supported integrity check value algorithms list.

Each SUPPORTED INTEGRITY CHECK VALUE ALGORITHM field indicates one supported algorithm for computing CbCS integrity check values. The values in the SUPPORTED INTEGRITY CHECK VALUE ALGORITHM fields are selected from the codes that table 79 (see 5.13.8) lists as integrity checking (i.e., AUTH) algorithms, except for AUTH_COMBINED.

The SUPPORTED D-H ALGORITHM LIST LENGTH field indicates the number of bytes that follow in the supported Diffie-Hellman (D-H) algorithms list.

Each SUPPORTED D-H ALGORITHM field indicates one supported algorithm for constructing CbCS shared keys. The values in the SUPPORTED D-H ALGORITHM fields are selected from the codes that table 79 (see 5.13.8) lists as Diffie-Hellman algorithms with finite field D-H computations.

The SUPPORTED CBCS METHODS LIST LENGTH field indicates the number of bytes that follow in the supported CbCS methods list.

Each SUPPORTED CBCS METHODS field indicates one supported CbCS method (see 6.r.2.3). The values in the SUPPORTED CBCS METHODS fields are selected from the codes listed in table x15 (see 6.r.2.3).

7.6.c.3.4 Security Token CbCS page

The Security Token CbCS page (see table x25) indicates the value of the security token (see x.x.x) for the I_T nexus on which the SECURITY PROTOCOL IN command was received.

Table x25 — Security Token CbCS page format

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)			PAGE CODE (003Fh)				
1								(LSB)
2	(MSB)			PAGE LENGTH (n-3)				
3								(LSB)
4	(MSB)			SECURITY TOKEN				
n								(LSB)

The PAGE CODE field shall be set to 003Fh to indicate that the Security Token CbCS page is being returned.

The page length field indicates the number of bytes that follow in the Security Token CbCS page.

The SECURITY TOKEN field shall contain the security token (see 5.13.6.8.10) for the I_T nexus on which the command was received.

7.6.c.3.5 Current CbCS Parameters CbCS page

The Current CbCS Parameters CbCS page (see table x26) indicates the current values for the CbCS parameters (see 5.13.6.8.15) used by the SCSI target device or logical unit as follows:

   a) If the logical unit to which the SECURITY PROTOCOL IN command is addressed is the SECURITY PROTOCOL well known logical unit (see 8.5), then the contents of the Current CbCS Parameters CbCS page apply to the SCSI target device; or

b) If the logical unit to which the SECURITY PROTOCOL IN command is addressed is not the SECURITY PROTOCOL well known logical unit, then the contents of the Current CbCS Parameters CbCS page apply to the addressed logical unit.

Table x26 — Current CbCS Parameters CbCS page format

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)			PAGE CODE (0040h)				
1								(LSB)
2	(MSB)			PAGE LENGTH (n-3)				
3								(LSB)
4				Reserved				
6								
7				MINIMUM ALLOWED CBCS METHOD				
8	(MSB)			POLICY ACCESS TAG				
11								(LSB)
12				Reserved				
14								
15	(MSB)			MASTER KEY IDENTIFIER				
22								(LSB)
23	(MSB)			WORKING KEY 0 IDENTIFIER				
30								(LSB)
31	(MSB)			WORKING KEY 1 IDENTIFIER				
38								(LSB)
				⋮				
144	(MSB)			WORKING KEY 15 IDENTIFIER				
151								(LSB)
152	(MSB)			CLOCK				
157								(LSB)
158				Reserved				
159				SECURITY TOKEN LENGTH (n-159)				
160	(MSB)			SECURITY TOKEN				
n								(LSB)

The PAGE CODE field shall be set to 0040h to indicate that the Current CbCS Parameters CbCS page is being returned.

The page length field indicates the number of bytes that follow in the Current CbCS Parameters CbCS page.

The MINIMUM ALLOWED CBCS METHOD field indicates the smallest value allowed in the CBCS METHOD field (see 6.r.2.3) of a capability descriptor processed by the SCSI target device or logical unit. The value in the MINIMUM ALLOWED CBCS METHOD field is selected from those listed in table x15 (see 6.r.2.3).

The SECURITY TOKEN LENGTH field indicates the number of bytes that follow in the SECURITY TOKEN field.

The SECURITY TOKEN field shall contain the security token (see 5.13.6.8.10) for the I_T nexus on which the command was received.

7.6.c.3.6 Set Master Key – Seed Exchange CbCS page

7.6.c.4 CbCS SECURITY PROTOCOL OUT CDB description

The CbCS SECURITY PROTOCOL OUT CDB has the format defined in 6.31 with the additional requirements described in this subclause.

When the SECURITY PROTOCOL field is set to CbCS (i.e., 07h) in a SECURITY PROTOCOL OUT command, the SECURITY PROTOCOL SPECIFIC field (see table x27) specifies the CbCS page to be returned in the parameter data (see 7.6.c.5). If the CBCS bit is set to one in the Extended INQUIRY Data VPD page (see 7.7.4), the CbCS SECURITY PROTOCOL IN command support requirements are shown in table x27.

Table x27 — SECURITY PROTOCOL SPECIFIC field for the CbCS SECURITY PROTOCOL OUT command

Code	CbCS page sent	Support	Reference
0000h – 0040h	Reserved		
0041h	Set Policy Access Tag	Optional	7.6.c.5.1
0042h	Set Minimum CbCS Method	Optional	7.6.c.5.2
0043h – 007Fh	Reserved		
0080h	Set Key	Mandatory	7.6.c.5.3
0081h – CFFFh	Reserved		
D000h	Set Master Key – Seed Exchange	Mandatory	7.6.c.5.4
D001h	Set Master Key – Change Master Key	Mandatory	7.6.c.5.5
D002h – FFFFh	Reserved		

If a CbCS SECURITY PROTOCOL OUT command is received with the INC_512 bit set to one, 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.

7.6.c.5 CbCS SECURITY PROTOCOL OUT parameter data

7.6.c.5.1 Set Policy Access Tag CbCS page

7.6.c.5.2 Set Minimum CbCS Method CbCS page

7.6.c.5.3 Set Key CbCS page

7.6.c.5.4 Set Master Key – Seed Exchange CbCS page

7.6.c.5.5 Set Master Key – Change Master Key CbCS page

…

7.7 Vital product data parameters

…

7.7.4 Extended INQUIRY Data VPD page

The Extended INQUIRY Data VPD page (see table 437) provides the application client with a means to obtain information about the logical unit.

Table 437 — Extended INQUIRY Data VPD page

Bit Byte	7	6	5	4	3	2	1	0
0	PERIPHERAL QUALIFIER			PERIPHERAL DEVICE TYPE				
1	PAGE CODE (86h)							
2	Reserved							
3	PAGE LENGTH (3Ch)							
4	Reserved		SPT			GRD_CHK	APP_CHK	REF_CHK
5	Reserved		UASK_SUP	GROUP_SUP	PRIOR_SUP	HEADSUP	ORDSUP	SIMPSUP
6	Reserved				WU_SUP	CRD_SUP	NV_SUP	V_SUP
7	Reserved							LUICLR
8	Reserved							CBCS
9 ~~8~~ 63	Reserved							

…

A capability-based command security (CBCS) bit set to one indicates that the logical unit supports the capability-based command security technique (see 5.13.6.8). A CBCS bit set to zero indicates that the logical unit does not support the capability-based command security technique.

7.7.5 Management Network Addresses VPD page

{{Several *pro forma* changes follow this note. The subclause and table numbering is reset at this point.}}

5.6 Reservations

…

5.6.1 Persistent Reservations overview

…

For each command, this standard or a command standard (see 3.1.18) defines the conditions that result in RESERVATION CONFLICT. Command standards define the conditions either in the device model or in the descriptions each of specific command.

Table 35 — SPC commands that are allowed in the presence of various reservations

| Command | Addressed logical unit has this type of persistent reservation held by another I_T nexus | | | | |
| | From any I_T nexus | | From registered I_T nexus (RR all types) | From not registered I_T nexus | |
	Write Excl	Excl Access		Write Excl RR	Excl Access – RR
…	…	…	…	…	…
RECEIVE COPY RESULTS	Conflict	Conflict	Allowed	Conflict	Conflict
RECEIVE CREDENTIAL	Conflict	Conflict	Allowed	Conflict	Conflict
…					

…

6.1 Summary of commands for all device types

The operation codes for commands that apply to all device types when the MCHNGR bit is set to zero, the SCCS bit is set to zero, and the ENCSERV bit is set to zero in the standard INQUIRY data (see 6.4.2) are listed in table 75.

<p align="center">Table 75 — Commands for all device types</p>

Command name	Operation code	Type	Reference
…	…	…	…
RECEIVE COPY RESULTS	84h	O	6.18
RECEIVE CREDENTIAL	7Fh/1800h ᵃ	O	6.r
Type Key: C = Command implementation is defined in the applicable command standard (see 3.1.18). M = Command implementation is mandatory. O = Command implementation is optional. Z = Command implementation is defined in a previous standard.			
ᵃ This command is defined by a combination of operation code and service action. The operation code value is shown preceding the slash and the service action value is shown after the slash.			

…

6.30 SECURITY PROTOCOL IN command

…

The SECURITY PROTOCOL field (see table 220) specifies which security protocol is being used.

<p align="center">Table 220 — SECURITY PROTOCOL field in SECURITY PROTOCOL IN command</p>

Code	Description	Reference
00h	Security protocol information	7.6.1
01h - 06h	Defined by the TCG	3.1.169
07h	CbCS	7.6.c
07h 08h - 1Fh	Reserved	
20h	Tape Data Encryption	SSC-3
21h	Data Encryption Configuration	TBD
22h - 3Fh	Reserved	
40h	SA Creation Capabilities	7.6.2
41h	IKEv2-SCSI	7.6.3
42h - ECh	Reserved	
EDh	SD Card TrustedFlash specification	3.1.137
EEh	Authentication in Host Attachments of Transient Storage Devices	IEEE 1667
EFh	ATA Device Server Password Security	SAT-2
F0h - FFh	Vendor Specific	

…

6.31 SECURITY PROTOCOL OUT command

…

The SECURITY PROTOCOL field (see table 222) specifies which security protocol is being used.

Table 222 — SECURITY PROTOCOL field in SECURITY PROTOCOL OUT command

Code	Description	Reference
00h	Reserved	
01h - 06h	Defined by the TCG	3.1.169
07h	CbCS	7.6.c
~~07h~~ 08h - 1Fh	Reserved	
20h	Tape Data Encryption	SSC-3
21h	Data Encryption Configuration	TBD
22h - 40h	Reserved	
41h	IKEv2-SCSI	7.6.3
42h - ECh	Reserved	
EDh	SD Card TrustedFlash specification	3.1.137
EEh	Authentication in Host Attachments of Transient Storage Devices	IEEE 1667
EFh	ATA Device Server Password Security	SAT-2
F0h - FFh	Vendor Specific	

…

D.3.5 Variable length CDB service action codes

The variable length CDB service action codes assigned by this standard are shown in table D.8.

Table D.8 — Variable Length CDB Service Action Codes Used by All Device Types

Service Action Code	Description
1800h	RECEIVE CREDENTIAL command
1801h - 1FFFh	Reserved
~~1800h - 1FFFh~~	~~Reserved~~