Date: October 18, 2007

To: T10 Committee (SCSI)

From: George Penokie (IBM)

Subject: SPC-4: UML model for CbCS

1 Overview

This proposal defines the UML model for CbCS. It will replace the current overview section of the SCSI commands standard proposal (07-069).

2 SPC-4 changes to add the Security Manager type

In section 6.4.2 (Standard INQUIRY data) add the security manager to the following table:

Table 1 — Peripheral device type

Code	Doc. [a]	Description
13h	SPC-4	Security Manager
14h - 1Dh		Reserved

[a] All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent editions of the listed standards.

[b] All well known logical units use the same peripheral device type code.

In section C.3.1 Operation codes add in a column for the security manager.

Editor's Note 1: All new section follows

---

9 Security manager command set

A security manager shall only process the commands listed in table 2. If a command is received by the security manager that is not listed in table 2, 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 COMMAND OPERATION CODE.

Table 2 — Commands for the security manager

Command name	Operation code	Type	Reference
INQUIRY	12h	M	6.4
REPORT LUNS	A0h	M	6.22
REQUEST SENSE	03h	M	6.28
RECEIVE CREDENTIAL	7Fh/1800h	M	x.xx
SECURITY PROTOCOL IN	A2h	M	6.29
SECURITY PROTOCOL OUT	B5h	M	6.30
TEST UNIT READY	00h	M	6.32

Key: M = Command implementation is mandatory.

5.13.5 Capability based Command Security

5.13.5.1 Overview

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

The CbCS protocol enables centralized management of SCSI command security.

CbCS secures access to a logical unit or a volume (see SSC-3) by providing cryptographic integrity of credentials that are added to commands sent to the logical unit (See ?2.7). This cryptographic integrity is based on mutual trust and key exchanges between the CbCS management device server, CbCS management application client, and the enforcement manager (see ?2.8).

Different levels of protection and security are achieved by using different CbCS methods. The following CbCS methods are defined by this standard:

a) The BASIC CbCS method (see x.x.x) provides protection against errors but does not prevent unauthorized access caused by means of malicious attacks (e.g., identity spoofing and network attacks); and
b) The CAPKEY CbCS method (see x.x.x) enforces application client authentication and provides cryptographic integrity of credentials. It protects against the following types of unauthorized access attacks:
   A) Illegal use of credentials beyond their original scope and lifespan;
   B) Forging or stealing credentials; and
   C) Using malformed credentials;

   Combined with a service delivery subsystem that has cryptographic message integrity, the CAPKEY CbCS method also protects against the following types of unauthorized access attacks:

   A) Network errors and malicious message modifications; and
   B) Message replay attacks.

CbCS also supports rapid revocation of credentials, per SCSI target device and per logical unit.

CbCS does not define task management function security.

CbCS (see figure 1) is composed of a:

a) Security Manager class (see 5.13.5.2) that contains:
   A) A CbCS Management Device Server class (see 5.13.5.3); and
   B) A CbCS Management Application Client class (see 5.13.5.4).
b) SCSI Initiator Device class (see SAM-4) that contains:
   A) A Secure CDB Originator class (see 5.13.5.5);

   and

c) SCSI Target Device class (see SAM-4) that contains:
   A) A Secure CDB Processor class (see 5.13.5.6):and
   B) An Enforcement Manager class (see 5.13.5.7);

Figure 1 shows the flow of transactions between the components of a CbCS capable SCSI domain.

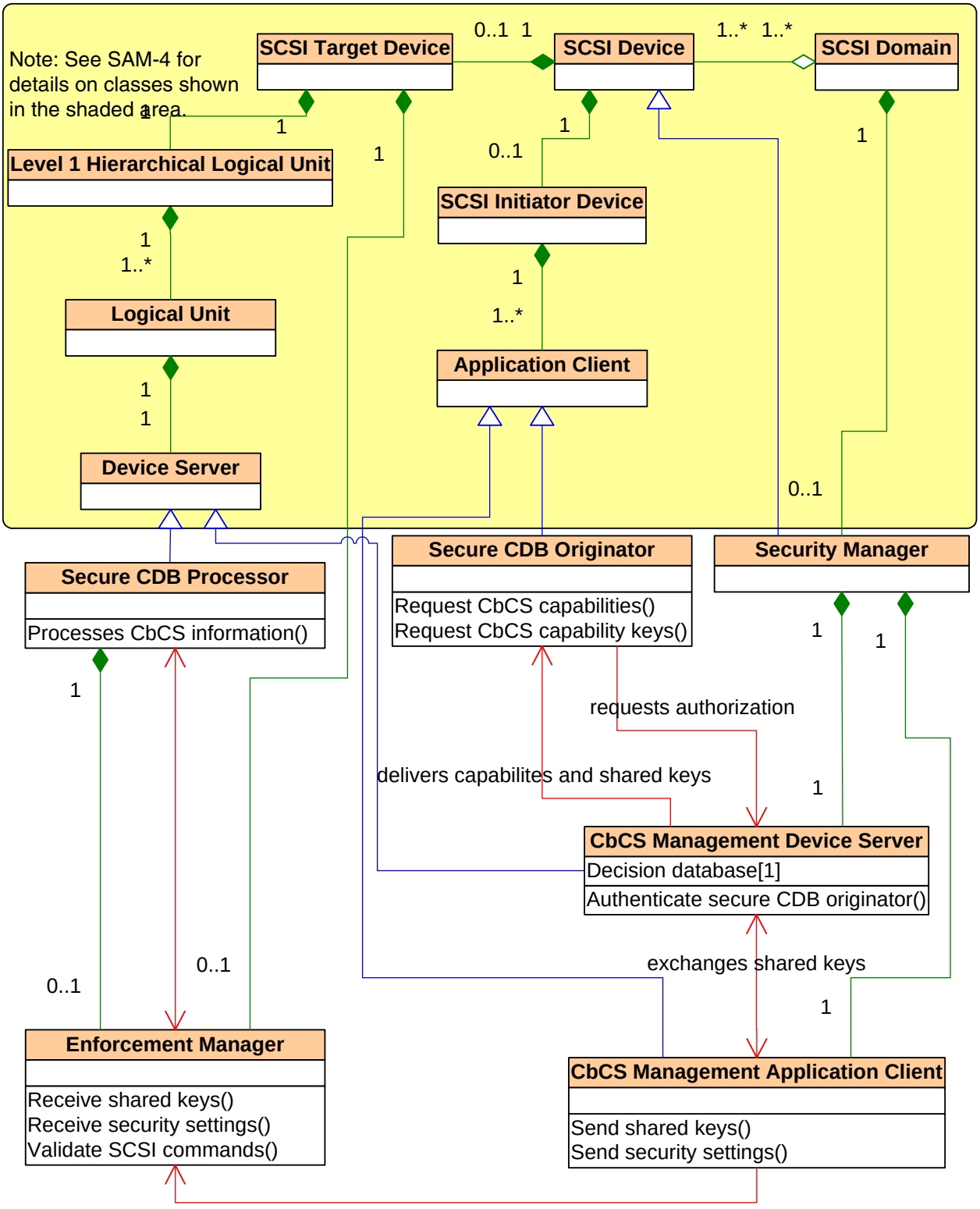

Figure 1 — CbCS overview class diagram

Each instance of CbCS shall contain:

a) one security manager that shall contain:
   A) one CbCS management device server; and

B) one CbCS management application.

b) one or more SCSI initiator devices that shall contain:
   A) one or more secure CDB originators;

   and

c) one or more SCSI target devices that shall contain:
   A) one secure CDB processor per logical unit; and
      a) one enforcement manager per secure CDB processor;
      b) one enforcement manager per SCSI target device; or
      c) both.

5.13.5.2 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 or volume (see SSC-3). It communicates with secure CDB originators to provide them CbCS credentials (see ?2.2.2.2), and with enforcement managers (see ?2.3) to provide them shared keys and security settings.

Editor's Note 2: The statement 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 or volume (see SSC-3). is a duplicate of a statement in the 07-262r2 proposal section 5.13.x.5 Security Manager class which states "The Security Manager class contains a decision database and a decision database update management mechanism whose definition is outside the scope of this standard and communicates with the Secure CDB Originator class (see 5.13.x.2) and the Enforcement Manager class (see 5.13.x.4) as shown in table x1." That statement should be modified to "The Security Manager class communicates with the Secure CDB Originator class (see 5.13.x.2) and the Enforcement Manager class (see 5.13.x.4) as shown in table x1."

The security manager may be located and may communicate with secure CDB originators and enforcement managers as follows:

a) If it is a SCSI device contained within the same SCSI domain as the secure CDB originator and the enforcement manager, it shall contain an application client and use it to communicate to the enforcement manager, and it shall contain a device server and use it to communicate with secure CDB originators;
b) If it is an application client located in the same device as the secure CDB originators, it shall communicate to the enforcement manager via the SCSI domain's service delivery subsystem, and it may communicate with the secure CDB originators by means outside the scope of this standard; and
c) If it is a device server located in the same device as the secure CDB processor, it shall communicate to the secure CDB originators via the SCSI domain's service delivery subsystem, and it may communicate with the enforcement manager by means outside the scope of this standard.

The security manager's device server is called CbCS management device server. The security manager's application client is called CbCS management application client (see ?2.2.3).

If the security manager is a SCSI device, it shall perform CbCS management using the CbCS management application client and the CbCS management device server as follows:

a) The CbCS management device server provides access policy controls to secure CDB originators using policy-coordinated CbCS capabilities; and
b) The CbCS management application client, in concert with the CbCS management device server, the Enforcement Manager, and the secure CDB processor prevents unsecured access to a logical unit or a volume (see SSC-3).

CbCS management is confined to the CbCS management application client and CbCS management device server. The communication of CbCS management information may occur in a manner outside the scope of this standard.

5.13.5.3 CbCS Management Device Server class

5.13.5.3.1 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).

The CbCS management device server shall authenticate the secure CDB originator unless the BASIC CbCS method is used (see ?2.6.2).

5.13.5.3.2 Decision Database attribute

The Decision Database attribute is used to obtain the authorization information required for deciding the type and duration of access granted to a secure CDB originator for a given logical unit or volume (see SSC-3) within a SCSI target device. CbCS Credentials are prepared by the CbCS management device server based on the contents of that Decision Database attribute.

5.13.5.4 CbCS Management Application Client class

The CbCS Management Application Client class sends shared keys and security settings to the enforcement manager using SECURITY PROTOCOL OUT commands and SECURITY PROTOCOL IN commands 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 ?6.1) the CbCS management application client may use to set and manage the shared keys stored in an enforcement manager.

5.13.5.5 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 or volume (see SSC-3). The secure CDB originator sends the CbCS capability (see xxx) and CbCS validation tag (see xxx) to the logical unit's secure CDB processor as part of a CbCS extended CDB.

For more information on the Secure CDB Originator class see 5.13.x.2.

5.13.5.6 Secure CDB Processor class

The Secure CDB Processor class:

a) Receives CbCS capability (see 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 ?4). If the CbCS bit is set to one, the logical unit shall support the following:

a) Extended SCSI command (see 4.3.4.2 [07-029r3]);
b) CbCS extension type (see ?3);
c) SECURITY PROTOCOL IN commands specifying the CbCS security protocol (see ?5.1); and
d) SECURITY PROTOCOL OUT commands specifying the CbCS security protocol (see ?6.1).

For more information on the Secure CDB Processor class see 5.13.x.3.

5.13.5.7 Enforcement Manager class

The Enforcement Manager class:

a) Receives shared keys (see ?2.8) and other security settings (see ?2.10) from the CbCS management application client;
b) Authenticates the CbCS capability (see x.x) with an integrity check value (see ?2.7.3, ?2.7.4) using the CbCS capability received from a secure CDB processor: and
c) Validates SCSI commands sent by secure CDB originators.

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 security settings it uses pertain to the logical unit (See ?2.10). If the enforcement manager is contained within the SCSI target device then, the shared keys and security settings it uses pertain to the SCSI target device, and the security protocol well-known logical unit is used for the commands to set shared keys and security settings (See ?6.1.).

If the shared key is stored in a well known logical unit then the 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.x.4.