SPECIFICATION

TCG Storage Security Subsystem Class: Opal

Version 2.02
Revision 1.0
Jan 24, 2022

Contact: admin@trustedcomputinggroup.org

PUBLISHED

DISCLAIMERS, NOTICES, AND LICENSE TERMS

THIS SPECIFICATION IS PROVIDED “AS IS” WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE.

Without limitation, TCG disclaims all liability, including liability for infringement of any proprietary rights, relating to use of information in this specification and to the implementation of this specification, and TCG disclaims all liability for cost of procurement of substitute goods or services, lost profits, loss of use, loss of data or any incidental, consequential, direct, indirect, or special damages, whether under contract, tort, warranty or otherwise, arising in any way out of use or reliance upon this specification or any information herein.

Contact the Trusted Computing Group at www.trustedcomputinggroup.org for information on specification licensing through membership agreements.

Any marks and brands contained herein are the property of their respective owners.

CONTENTS

DISCLAIMERS, NOTICES, AND LICENSE TERMS 1

TABLES........ 5

1 Introduction........ 6

1.1 Document Purpose 6

1.2 Scope and Intended Audience 6

1.3 Key Words 6

1.3.1 Fonts........ 6

1.4 Statement Type........ 6

1.5 Document References 7

1.6 Document Precedence........ 7

1.7 Opal SSC Terminology 7

1.8 Legend........ 8

2 Opal SSC Overview........ 9

2.1 Opal SSC Use Cases and Threats........ 9

2.2 Security Providers (SPs) 9

2.3 Interface Communication Protocol 9

2.4 Cryptographic Features........ 9

2.5 Authentication 9

2.6 Table Management 10

2.7 Access Control & Personalization 10

2.8 Issuance........ 10

2.9 SSC Discovery........ 10

2.10 Mandatory Feature Sets 10

3 Opal SSC Features........ 11

3.1 Security Protocol 1 Support 11

3.1.1 Level 0 Discovery (M) 11

3.2 Security Protocol 2 Support 22

3.2.1 ComID Management........ 22

3.2.2 Stack Protocol Reset (M) 22

3.2.3 TPER_RESET command (M) 22

3.3 Communications 23

3.3.1 Communication Properties........ 23

3.3.2 Supported Security Protocols........ 23

3.3.3 ComIDs........ 23

3.3.4 Synchronous Protocol........ 24

3.3.5 Storage Device Resets 25
3.3.6 Protocol Stack Reset Commands (M) 26
4 Opal SSC-compliant Functions and SPs 27
4.1 Session Manager 27
4.1.1 Methods 27
4.2 Admin SP 28
4.2.1 Base Template Tables 29
4.2.2 Base Template Methods 41
4.2.3 Admin Template Tables 41
4.2.4 Admin Template Methods 43
4.2.5 Opal Additional Column Types 43
4.2.6 Opal Additional Data Structures 43
4.2.7 Opal Additional Tables 44
4.2.8 Crypto Template Tables 44
4.2.9 Crypto Template Methods 44
4.3 Locking SP 44
4.3.1 Base Template Tables 44
4.3.2 Base Template Methods 76
4.3.3 Crypto Template Tables 76
4.3.4 Crypto Template Methods 76
4.3.5 Locking Template Tables 76
4.3.6 Locking Template Methods 81
4.3.7 SD Read/Write Data Command Locking Behavior Interactions with Range Crossing 81
4.3.8 Non Template Tables 81
5 Appendix – SSC Specific Features 82
5.1 Opal SSC-Specific Methods 82
5.1.1 Activate – Admin Template SP Object Method 82
5.1.2 Revert – Admin Template SP Object Method 82
5.1.3 RevertSP – Base Template SP Method 84
5.2 Life Cycle 86
5.2.1 Issued vs. Manufactured SPs 86
5.2.2 Manufactured SP Life Cycle States 86
5.2.3 Type Table Modification 88
5.3 Byte Table Access Granularity 89
5.3.1 Table Table Modification 89
5.4 Examples of Alignment Geometry Reporting 91

TABLES

Table 1 Opal SSC Terminology 7
Table 2 SP Tables Legend 8
Table 3 Level 0 Discovery Header 11
Table 4 Level 0 Discovery - TPer Feature Descriptor 12
Table 5 Level 0 Discovery - Locking Feature Descriptor 13
Table 6 Level 0 Discovery - Geometry Reporting Feature Descriptor 14
Table 7 Level 0 Discovery - Opal SSC V2 Feature Descriptor 16
Table 8 - SSC Minor Versions 17
Table 9 Level 0 Discovery – Supported Data Removal Mechanism Feature Descriptor 18
Table 10 Parameter explanation 19
Table 11 Supported Data Removal Mechanism 21
Table 12 Data Removal Time (Data Removal Time Format bit= 0) 22
Table 13 Data Removal Time (Data Removal Time Format bit= 1) 22
Table 14 TPER_RESET Command 23
Table 15 ComID Assignments 24
Table 16 Supported Tokens 24
Table 17 reset_types 26
Table 18 Properties Requirements 27
Table 19 Admin SP - SPInfo Table Preconfiguration 29
Table 20 Admin SP - SPTemplates Table Preconfiguration 29
Table 21 Admin SP - Table Table Preconfiguration 29
Table 22 Admin SP - MethodID Table Preconfiguration 31
Table 23 Admin SP - AccessControl Table Preconfiguration 32
Table 24 Admin SP - ACE Table Preconfiguration 39
Table 25 Admin SP - Authority Table Preconfiguration 40
Table 26 Admin SP - C_PIN Table Preconfiguration 40
Table 27 Admin SP – TPerInfo Columns 41
Table 28 Admin SP - TPerInfo Table Preconfiguration 42
Table 29 Admin SP - Template Table Preconfiguration 42
Table 30 Admin SP - SP Table Preconfiguration 42
Table 31 data_removal_mechanism Type Table Addition 43
Table 32 data_removal_mechanism Enumeration Values 43
Table 33 DataRemovalMechansim Table Description 43
Table 34 Admin SP – DataRemovalMechansim Table Preconfiguration 44
Table 35 Locking SP - SPInfo Table Preconfiguration 44
Table 36 Locking SP - SPTemplates Table Preconfiguration 45
Table 37 Locking SP - Table Table Preconfiguration 45
Table 38 Locking SP - MethodID Table Preconfiguration 46
Table 39 Locking SP - AccessControl Table Preconfiguration 48
Table 40 Locking SP - ACE Table Preconfiguration 70
Table 41 Locking SP - Authority Table Preconfiguration 74
Table 42 Locking SP - C_PIN Table Preconfiguration 75
Table 43 Locking SP - SecretProtect Table Preconfiguration 76
Table 44 Locking SP – LockingInfo Columns 77
Table 45 Locking SP - LockingInfo Table Preconfiguration 77
Table 46 Locking SP - Locking Table Preconfiguration 78
Table 47 Locking SP - MBRControl Table Preconfiguration 80
Table 48 Locking SP - K_AES_128 Table Preconfiguration 80
Table 49 Locking SP - K_AES_256 Table Preconfiguration 81
Table 50 LifeCycle Type Table Modification 88
Table 51 Table Table Additional Columns 89

1 Introduction

1.1 Document Purpose

Storage Workgroup specifications provide a comprehensive architecture for putting Storage Devices under policy control as determined by the trusted platform host, the capabilities of the Storage Device to conform to the policies of the trusted platform, and the lifecycle state of the Storage Device as a Trusted Peripheral.

1.2 Scope and Intended Audience

This specification defines the Opal Security Subsystem Class (SSC). Any SD that claims Opal SSC compatibility SHALL conform to this specification.

The intended audience for this specification is both trusted Storage Device manufacturers and developers that want to use these Storage Devices in their systems.

1.3 Key Words

Key words are used to signify SSC requirements.

The Key Words "**SHALL**", "**SHALL NOT**", "**SHOULD**," and "**MAY**" are used in this document. These words are a subset of the RFC 2119 key words used by TCG, and have been chosen since they map to key words used in T10/T13 specifications. These key words are to be interpreted as described in [1].

In addition to the above key words, the following are also used in this document to describe the requirements of particular features, including tables, methods, and usages thereof.

- **Mandatory (M):** When a feature is Mandatory, the feature SHALL be implemented. A Compliance test SHALL validate that the feature is operational.
- **Optional (O):** When a feature is Optional, the feature MAY be implemented. If implemented, a Compliance test SHALL validate that the feature is operational.
- **Excluded (X):** When a feature is Excluded, the feature SHALL NOT be implemented. A Compliance test SHALL validate that the feature is not operational.
- **Not Required (N)** When a feature is Not Required, the feature MAY be implemented. No Compliance test is required.

1.3.1 Fonts

Names of methods and SP tables are in Courier New font (e.g., the Set method, the Locking table). This convention does not apply to method and table names appearing in headings or captions.

1.4 Statement Type

Please note a very important distinction between different sections of text throughout this document. There are two distinctive kinds of text: informative comment and normative statements. Because most of the text in this specification will be of the kind normative statements, the authors have informally defined it as the default and, as such, have specifically called out text of the kind informative comment. They have done this by flagging the beginning and end of each informative comment and highlighting its text in gray. This means that unless text is specifically marked as of the kind informative comment, it can be considered a kind of normative statements.

EXAMPLE: Start of Informative Comment

This is the first paragraph of 1–n paragraphs containing text of the kind *informative comment* ...

This is the second paragraph of text of the kind *informative comment* ...

This is the nth paragraph of text of the kind *informative comment* ...

To understand the TCG specification the user must read the specification. (This use of MUST does not require any action).

End of Informative Comment

1.5 Document References

[1]. IETF RFC 2119, 1997, "Key words for use in RFCs to Indicate Requirement Levels"

[2]. Trusted Computing Group (TCG), "TCG Storage Architecture Core Specification", Version 2.01

[3]. NIST, FIPS-197, 2001, "Advanced Encryption Standard (AES)"

[4]. Trusted Computing Group (TCG), "TCG Storage Interface Interactions Specification", Version 1.09

[5]. Trusted Computing Group (TCG), "TCG Storage Security Subsystem Class: Opal", Versions 1.00, 2.00

[6]. Trusted Computing Group (TCG), "TCG Storage Opal SSC Feature Set: Additional DataStore Tables", Version 1.00

[7]. Trusted Computing Group (TCG), "TCG Storage Opal SSC Feature Set: PSID", Version 1.00

[8]. Trusted Computing Group (TCG), "TCG Storage Feature Set: Block SID Authentication", Version 1.00

1.6 Document Precedence

In the event of conflicting information in this specification and other documents, the precedence for requirements is:

1. This specification
2. Storage Interface Interactions Specification [4]
3. TCG Storage Architecture Core Specification [2]

1.7 Opal SSC Terminology

Table 1 provides special definitions that are not defined in [2].

Table 1 Opal SSC Terminology

Term	Definition
SD	Storage Device
SP	Security Provider
TPer	Trusted Peripheral
Manufactured SP	A Manufactured SP is an SP that was created and preconfigured during the SD manufacturing process
N/A	Not Applicable.
Original Factory State (OFS)	The original state of an SP when it was created in manufacturing, including its table data, access control settings, and life cycle state. Each Manufactured SP has its own Original Factory State. Original Factory State applies to Manufactured SPs only.
Vendor Unique (VU)	These values are unique to each SD manufacturer. Typically VU is used in table cells.
Preconfiguration Data	The default data in the OFS.

Term	Definition
MM MM	The LSBs of a User Authority object's UID (hexadecimal) as well as the corresponding `C_PIN` credential object's UID (hexadecimal)
NN NN	The LSBs of a Locking object's UID (hexadecimal) as well as the corresponding K_AES_128/K_AES_256 object's UID (hexadecimal)
XX XX	The LSBs of an Admin Authority object's UID (hexadecimal) as well as the corresponding `C_PIN` credential object's UID (hexadecimal)

1.8 Legend

The legends in Table 2 define SP table cell color coding. This color coding is informative only. The table cell content is normative.

Table 2 SP Tables Legend

Table Cell Legend	R-W	Value	Access Control	Comment
Arial-Narrow	Read-only	Opal SSC specified	Fixed	• Cell content is Read-Only. • Access control is fixed. • Value is specified by the Opal SSC
Arial Narrow bold-under	Read-only	VU	Fixed	• Cell content is Read-Only. • Access Control is fixed. • Values are Vendor Unique (VU). A minimum or maximum value may be specified.
Arial-Narrow	Not Defined	(N)	Not Defined	• Cell content is (N). • Access control is not defined. • Any text in table cell is informative only. • A `Get` MAY omit this column from the method response.
Arial Narrow bold-under	Write	Preconfigured, user (i.e. host) personalizable	Preconfigured, user (i.e. host) personalizable	• Cell content is writable. • Access control is personalizable • `Get` Access Control is not described by this color coding
Arial-Narrow	Write	Preconfigured, user (i.e. host) personalizable	Fixed	• Cell content is writable. • Access control is fixed. • `Get` Access Control is not described by this color coding

2 Opal SSC Overview

2.1 Opal SSC Use Cases and Threats

Start of Informative Comment

The Opal SSC is an implementation profile for Storage Devices built to:

- Protect the confidentiality of stored user data against unauthorized access once it leaves the owner's control (following a power cycle and subsequent deauthentication)
- Enable interoperability between multiple SD vendors

An Opal SSC compliant SD:

- Facilitates feature discoverability
- Provides some user definable features (e.g. access control, locking ranges, user passwords, etc.)
- Supports Opal SSC unique behaviors (e.g. communication, table management)

This specification addresses a limited set of use cases. They are:

- Deploy Storage Device & Take Ownership: the Storage Device is integrated into its target system and ownership transferred by setting or changing the Storage Device's owner credential.
- Activate or Enroll Storage Device: LBA ranges are configured and data encryption and access control credentials (re)generated and/or set on the Storage Device. Access control is configured for LBA range unlocking.
- Lock & Unlock Storage Device: unlocking of one or more LBA ranges by the host and locking of those ranges under host control via either an explicit lock or implicit lock triggered by a reset event. MBR shadowing provides a mechanism to boot into a secure pre-boot authentication environment to handle device unlocking.
- Repurpose & End-of-Life: erasure of data within one or more LBA ranges and reset of locking credential(s) for Storage Device repurposing or decommissioning.

End of Informative Comment

2.2 Security Providers (SPs)

An Opal SSC compliant SD SHALL support at least two Security Providers (SPs):

1) Admin SP
2) Locking SP

The Locking SP MAY be created by the SD manufacturer.

2.3 Interface Communication Protocol

An Opal SSC compliant SD SHALL implement the synchronous communications protocol as defined in Section 3.3.4.

This communication protocol operates based upon configuration information defined by:

1) the values reported via Level 0 Discovery (see section 3.1.1);

The combination of the host's communication properties and the TPer's communication properties (see section 4.1.1.1).

2.4 Cryptographic Features

An Opal SSC compliant SD SHALL implement Full Disk Encryption for all host accessible user data stored on media. AES-128 or AES-256 SHALL be supported (see [3]).

2.5 Authentication

An Opal SSC compliant SD SHALL support password authorities and authentication.

2.6 Table Management

This specification defines the mandatory tables and mandatory/optional table rows delivered by the SD manufacturer. The creation or deletion of tables after manufacturing is outside the scope of this specification. The creation or deletion of table rows post-manufacturing is outside the scope of this specification.

2.7 Access Control & Personalization

Initial access control policies are preconfigured at SD manufacturing time on manufacturer created SPs. An Opal SSC compliant SD SHALL support personalization of certain Access Control Elements of the Locking SP.

2.8 Issuance

The Locking SP MAY be present in the SD when the SD leaves the manufacturer. The issuance of SPs is outside the scope of this specification.

2.9 SSC Discovery

Refer to [2] for details (see section 3.1.1).

2.10 Mandatory Feature Sets

An Opal SSC compliant SD SHALL support the following TCG Storage Feature Sets:

1) Additional DataStore Tables, Opal SSC Feature Set (refer to [6]);
2) PSID, Opal SSC Feature Set (refer to [6]).
3) Block SID Authentication Feature Set (refer to [8])

3 Opal SSC Features

3.1 Security Protocol 1 Support

3.1.1 Level 0 Discovery (M)

Refer to [2] for more details.

An Opal SSC compliant SD SHALL return the following Level 0 response:

- Level 0 Discovery Header (see Table 3)
- TPer Feature Descriptor (see Table 4)
- Locking Feature Descriptor (see Table 5)
- Opal SSC V2 Feature Descriptor (see Table 7)

Additionally, an Opal SSC compliant SD MAY return the following Level 0 response:

- Geometry Reporting Feature (see Table 6)
- Supported Data Removal Mechanism Feature Descriptor (see Table 9)

3.1.1.1 Level 0 Discovery Header

Table 3 Level 0 Discovery Header

Bit **Byte**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0	(MSB)	Length of Parameter Data						
1								
2								
3								(LSB)
4	(MSB)	Data structure revision						
5								
6								
7								(LSB)
8	(MSB)	Reserved						
…								
15								(LSB)
16	(MSB)	Vendor Specific						
…								
47								(LSB)

An Opal SSC compliant SD SHALL return the following:

- Length of parameter data = VU
- Data structure revision = 0x00000001 or
  any version that supports the defined features in this SSC
- Vendor Specific = VU

3.1.1.2 TPer Feature (Feature Code = 0x0001)

Table 4 Level 0 Discovery - TPer Feature Descriptor

Bit **Byte**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0	(MSB)	Feature Code (0x0001)						
1								(LSB)
2	Version				Reserved			
3	Length							
4	Reserved	ComID Mgmt Supported	Reserved	Streaming Supported	Buffer Mgmt Supported	ACK/NAK Supported	Async Supported	Sync Supported
5 - 15	Reserved							

An Opal SSC compliant SD SHALL return the following:

- Feature Code = 0x0001
- Version = 0x1 or any version that supports the defined features in this SSC
- Length = 0x0C
- ComID Mgmt Supported = VU
- Streaming Supported = 1
- Buffer Mgmt Supported = VU
- ACK/NACK Supported = VU
- Async Supported = VU
- Sync Supported = 1

3.1.1.3 Locking Feature (Feature Code = 0x0002)

** means the present current state of the respective feature

Table 5 Level 0 Discovery - Locking Feature Descriptor

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)	Feature Code (0x0002)						
1								(LSB)
2	Version				Reserved			
3	Length							
4	HW Reset for LOR/DOR Supported	MBR Shadowing Not Supported	MBR Done	MBR Enabled	Media Encryption	Locked	Locking Enabled	Locking Supported
5 - 15	Reserved							

An Opal SSC compliant Storage Device SHALL return the following:

- Feature Code = 0x0002
- Version = 0x3 or any version that supports the defined features in this SSC
- Length = 0x0C
- HW Reset for LOR/DOR Supported = VU
- MBR Shadowing Not Supported = 0
  - If MBR Shadowing feature is not absent (i.e., is supported), then this bit SHALL be 0.
  - If MBR Shadowing feature is absent (i.e., is not supported), then this bit SHALL be 1.
- MBR Done = **
- MBR Enabled = **
- Media Encryption = 1
- Locked = **
- Locking Enabled = See section 3.1.1.3.1
- Locking Supported = 1

3.1.1.3.1 LockingEnabled Definition

The definition of the LockingEnabled bit is changed from [2] as follows:

The LockingEnabled bit SHALL be set to one if an SP that incorporates the Locking template is in any state other than Nonexistent or Manufactured-Inactive; otherwise, the LockingEnabled bit SHALL be set to zero.

3.1.1.4 Geometry Reporting Feature (Feature Code = 0x0003)

3.1.1.4.1 Overview

This information indicates support for logical block and physical block geometry. This feature MAY be returned in the Level 0 Discovery response. See [2] for additional information.

Table 6 Level 0 Discovery - Geometry Reporting Feature Descriptor

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)	Feature Code (0x0003)						
1								(LSB)
2	Version				Reserved			
3	Length							
4	Reserved							ALIGN
5		Reserved						
6								
7								
8								
9								
10								
11								
12	(MSB)	LogicalBlockSize						
13								
14								
15								(LSB)
16	(MSB)	AlignmentGranularity						
17								
18								
19								
20								
21								
22								
23								(LSB)
24	(MSB)	LowestAlignedLBA						
25								
26								

Bit Byte	7	6	5	4	3	2	1	0
27								
28								
29								
30								
31								(LSB)

An Opal SSC compliant SD SHALL return the following:

- Feature Code = 0x0003
- Version = 0x01
- Length = 0x1C

3.1.1.4.2 ALIGN

If the value of the AlignmentRequired column of the `LockingInfo` table is TRUE, then the ALIGN bit shall be set to one. If the value of the AlignmentRequired column of the `LockingInfo` table is FALSE, then the ALIGN bit shall be cleared to zero.

3.1.1.4.3 LogicalBlockSize

LogicalBlockSize SHALL be set to the value of the LogicalBlockSize column in the `LockingInfo` table.

3.1.1.4.4 AlignmentGranularity

AlignmentGranularity SHALL be set to the value of the AlignmentGranularity column in the `LockingInfo` table.

3.1.1.4.5 LowestAlignedLBA

LowestAlignedLBA SHALL be set to the value of the LowestAlignedLBA column in the `LockingInfo` table.

3.1.1.5 Opal SSC V2 Feature (Feature Code = 0x0203)

Table 7 Level 0 Discovery - Opal SSC V2 Feature Descriptor

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)	Feature Code (0x0203)						
1								(LSB)
2	Feature Descriptor Version Number				SSC Minor Version Number			
3	Length							
4	(MSB)	Base ComID						
5								(LSB)
6	(MSB)	Number of ComIDs						
7								(LSB)
8	Reserved for future common SSC parameters							Range Crossing Behavior
9	(MSB)	Number of Locking SP Admin Authorities Supported						
10								(LSB)
11	(MSB)	Number of Locking SP User Authorities Supported						
12								(LSB)
13	Initial C_PIN_SID PIN Indicator							
14	Behavior of C_PIN_SID PIN upon TPer Revert							
15-19	Reserved for future common SSC parameters							

An Opal SSC compliant Storage Device SHALL return the following:

- Feature Code = 0x0203
- Feature Descriptor Version Number = 0x2 or any version that supports the defined features in this SSC
- SSC Minor Version Number = As specified in Table 8
- Length = 0x10
- Base ComID = VU
- Number of ComIDs = 0x0001 or larger
- Range Crossing Behavior = VU
  - 0 = The Storage Device supports commands addressing consecutive LBAs in more than one LBA range if all the LBA ranges addressed are unlocked. See section 4.3.7.
  - 1 = The Storage Device terminates commands addressing consecutive LBAs in more than one LBA range. See 4.3.7
- Number of Locking SP Admin Authorities = 4 or larger
- Number of Locking SP User Authorities = 8 or larger
- Initial C_PIN_SID PIN Indicator = VU

  - 0x00 = The initial C_PIN_SID PIN value is equal to the C_PIN_MSID PIN value
  - 0xFF = The initial C_PIN_SID PIN value is VU, and MAY not be equal to the C_PIN_MSID PIN value
  - 0x01 – 0xFE = Reserved
- Behavior of C_PIN_SID PIN upon TPer Revert = VU
  - 0x00 = The C_PIN_SID PIN value becomes the value of the C_PIN_MSID `PIN` column after successful invocation of Revert on the Admin SP's object in the `SP` table
  - 0xFF = The C_PIN_SID PIN value changes to a VU value after successful invocation of Revert on the Admin SP's object in the `SP` table, and MAY not be equal to the C_PIN_MSID PIN value
  - 0x01 – 0xFE = Reserved

Table 8 - SSC Minor Versions

Opal Minor Version	Standard Referenced
0x00	TCG Opal SSC Specification v2.00
0x01	TCG Opal SSC Specification v2.01
0x02	TCG Opal SSC Specification v2.02
All others	Reserved

If an Opal v2.00 SSC implementation is backward compatible with Opal v1.00, then the SD SHALL also report the Opal SSC Feature Descriptor as defined in [5].

Start of Informative Comment

An Opal v2.00 implementation is backward compatible to Opal v1.00 only if the geometry reported by the Geometry Reporting Feature does not specify any alignment restrictions (i.e. ALIGN = FALSE, see section 3.1.1.4.2) , and if the TPer does not specify any granularity restrictions for byte tables (i.e. MandatoryWriteGranularity = 1 for all byte tables, see section 5.3.1.1), and if the "Initial C_PIN_SID PIN Indicator" and "Behavior of C_PIN_SID PIN upon TPer Revert" fields are both 0x00.

End of Informative Comment

3.1.1.6 Supported Data Removal Mechanism Feature (Feature Code = 0x0404)

Table 9 Level 0 Discovery – Supported Data Removal Mechanism Feature Descriptor

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)	Feature Code (0x0404)						
1								(LSB)
2	Version				Reserved			
3	Length							
4	Reserved							
5	Reserved						Data Removal Operation Interrupted	Data Removal Operation Processing
6	Supported Data Removal Mechanism							
7	Reserved		Data Removal Time Format for Bit 5	Reserved		Data Removal Time Format for Bit 2	Data Removal Time Format for Bit 1	Data Removal Time Format for Bit 0
8-9	Data Removal Time for Supported Data Removal Mechanism Bit 0							
10-11	Data Removal Time for Supported Data Removal Mechanism Bit 1							
12-13	Data Removal Time for Supported Data Removal Mechanism Bit 2							
14-17	Reserved							
18-19	Data Removal Time for Supported Data Removal Mechanism Bit 5							
20-35	Reserved for future Supported Data Removal Mechanism parameters							

An Opal Compliant SD SHALL return the parameters listed in Table 10:

Table 10 Parameter explanation

Parameter	**Value**	**Details**
Feature code	0x0404	Feature code value
Version	0x02	Version of the descriptor
Length	0x20	Length of the feature descriptor
Data Removal Operation Processing		see section 3.1.1.6.1
Data Removal Operation Interrupted		see section 3.1.1.6.2
Reserved		Return all zeros
Supported Data Removal Mechanism		see section 3.1.1.6.3
Data Removal Time Format for each bit		see section 3.1.1.6.4

3.1.1.6.1 Data Removal Operation Processing Definition

The Data Removal Operation Processing bit SHALL be set to one if the TPer is performing any supported data removal operation including:

- `Revert`,
- `RevertSP`, or
- `GenKey`.

Otherwise, the Data Removal Operation Processing bit SHALL be set to zero. If the operation is in progress, the security transport commands such as the security send, and the security receive SHALL be processed by the SD. The Data Removal Operation Processing bit SHALL be set to zero upon a successful completion of a data removal operation.

The Data Removal Operation Processing bit SHALL be set to one if the data removal operation is restarted after a Power Cycle (see Table 17).

3.1.1.6.2 Data Removal Operation Interrupted

The Data Removal Operation Interrupted bit SHALL be set to one if a previously issued data removal operation such as `Revert`, `RevertSP` or `GenKey` was interrupted for any reason (including, power loss, interface reset, etc.). The Data Removal Operation Interrupted bit SHALL be set to zero after successful completion of a data removal operation.

Start of Informative Comment

The host can reissue a data removal operation that was interrupted (such as `RevertSP`, `Revert`, or `GenKey`), The SD can be in a locked state if the operation was interrupted and the SD is now operational.

End of Informative Comment

3.1.1.6.3 Supported Data Removal Mechanism Definition

Each bit of the Supported Data Removal Mechanism (see Table 11) SHALL be set to one if the TPer supports the corresponding Data Removal Mechanism; otherwise, each bit SHALL be set to zero. The TPer SHALL support the Crypto Erase mechanism and MAY support the Overwrite Data Erase or Block Erase or other mechanisms. The TPer MAY support multiple Data Removal Mechanisms described in Table 11. After a `RevertSP` method has completed without an error, the condition of user data SHALL be indicated as specified in Table 11.

Table 11 Supported Data Removal Mechanism

Bit	Name	Condition of user data after Data Removal
0	Overwrite Data Erase[1]	The Overwrite Data Erase mechanism causes TPer to alter information by writing a vendor specific data pattern to the medium.
1	Block Erase[1]	The Block Erase mechanism causes the TPer to alter information by setting the physical blocks to a vendor specific value.
2	Cryptographic Erase [2]	The TPer SHALL support this data erasure mechanism. Further this mechanism SHALL be executed in addition to any other supported data removal mechanism that is being executed. This bit MAY be used by the `Revert` or the `RevertSP` or the `GenKey` (band erase) mechanisms of data removal, where the cryptographic keys used to encrypt the user data are changed.
3-4	Reserved	Reserved
5	Vendor Specific Erase[1]	The Vendor Specific Erase mechanisms cause all user data to be removed by a vendor specific method. [3]
6-7	Reserved	

Notes:

[1] The cryptographic erase operation SHALL also be performed when any of the other data removal mechanisms are used.

[2] The Cryptographic Erase bit may be used by the `Revert` or the `RevertSP` or the `GenKey` operations (band erase). Any subsequent operation(s) such as Deallocate, or Unmap, or Trim, that is part of the implementation of the data removal operation SHALL be accounted for in the time reported for this operation (see section 3.1.1.6.4). The time value reported SHALL correspond to the estimated completion time of the Cryptographic Erase. For the erase (`GenKey`) operation, the reported estimated time value will correspond to the estimated completion time of the erase operation, regardless of the extent of the range being erased.

[3] If a SD supports more than one vendor proprietary method of data removal, then the associated estimated time value will represent the completion time for the longest vendor specific erase mechanism of data removal, then the associated estimated time value will represent the completion time for the longest of the vendor specific mechanisms.

3.1.1.6.4 Data Removal Time Format and Data Removal Time Definition

Each Data Removal Time field provides the worst case estimate of the time required to perform the erasure corresponding to each Data Removal Mechanism defined in the Supported Data Removal Mechanism field. The Data Removal Time Format bit identifies the format used to express the time as follows:

a) if the Data Removal Time Format bit is set to zero, then the estimated time is defined in Table 12; and

b) if the Data Removal Time Format bit is set to one, then the estimated time is defined in Table 13.

The Data Removal Time Format bit and Data Removal Time Format field are defined in Table 12 and Table 13.

Table 12 Data Removal Time (Data Removal Time Format bit= 0)

Value	**Time**
0	Not reported
1..65534	(Value x 2) seconds
65535	>= 131068 seconds

Table 13 Data Removal Time (Data Removal Time Format bit= 1)

Value	**Time**
0	Not reported
1..65534	(Value x 2) minutes
65535	>= 131068 minutes

Start of Informative Comment

Each Data Removal Time field gives an estimate of the total time required to perform the erasure for each corresponding Data Removal Mechanism. This field is not a dynamic estimate of the remaining time for completion.

When `GenKey` is performed on a range that's less than the global range, the time needed for the completion of the operation can be less than the time reported for the operation. The reported estimated time for the data removal operation will be for the entire capacity of the SD. The host software can use the ratio of the band size to the entire capacity of the SD, to derive the estimated time for erasing a band.

End of Informative Comment

3.2 Security Protocol 2 Support

3.2.1 ComID Management

ComID management support is reported in Level 0 Discovery. Statically allocated ComIDs are also discoverable via the Level 0 Discovery response.

3.2.2 Stack Protocol Reset (M)

An Opal SSC compliant SD SHALL support the Stack Protocol Reset command. Refer to [2] for details.

3.2.3 TPER_RESET command (M)

If the TPER_RESET command is enabled, it SHALL cause the following before the TPer accepts the next IF-SEND or IF-RECV command:

a) all dynamically allocated ComIDs SHALL return to the Inactive state;
b) all open sessions SHALL be aborted on all ComIDs;
c) all uncommitted transactions SHALL be aborted on all ComIDs;
d) the synchronous protocol stack for all ComIDs SHALL be reset to its initial state
e) all TCG command and response buffers SHALL be invalidated for all ComIDs;
f) all related method processing occurring on all ComIDs SHALL be aborted;

g) The TPer's knowledge of the host's communications capabilities, on all ComIDs, SHALL be reset to the initial minimum assumptions defined in [2] or the TPer's SSC definition;
h) the values of the ReadLocked and WriteLocked columns SHALL be set to True for all Locking SP's Locking objects that contain the Programmatic enumeration value in the LockedOnReset column;
i) the value of the Done column of the Locking SP's `MBRControl` table SHALL be set to False, if the DoneOnReset column contains the Programmatic enumeration value.

The TPER_RESET command is delivered by the transport IF-SEND command. If the TPER_RESET command is enabled, the TPer SHALL accept and acknowledge it at the interface level. If the TPER_RESET command is disabled, the TPer SHALL abort it at the interface level with the "Other Invalid Command Parameter" status (see [4]). There is no IF-RECV response to the TPER_RESET command.

The TPER_RESET command is defined in Table 14.

The Transfer Length SHALL be non-zero. All data transferred SHALL be ignored.

Table 14 TPER_RESET Command

FIELD	VALUE
Command	IF-SEND
Protocol ID	0x02
Transfer Length	Non-zero
ComID	0x0004

3.3 Communications

3.3.1 Communication Properties

The TPer SHALL support the minimum communication buffer size as defined in section 4.1.1.1. For each ComID, the physical buffer size SHALL be reported to the host via the `Properties` method.

The TPer SHALL terminate any IF-SEND command whose transfer length is greater than the reported MaxComPacketSize size for the corresponding ComID. For details, refer to "Invalid Transfer Length parameter on IF-SEND" in [4].

Data generated in response to methods contained within an IF-SEND command payload subpacket (including the required ComPacket / Packet / Subpacket overhead data) SHALL fit entirely within the response buffer. If the method response and its associated protocol overhead do not fit completely within the response buffer, the TPer

1) SHALL terminate processing of the IF-SEND command payload,
2) SHALL NOT return any part of the method response if the Sync Protocol is being used, and
3) SHALL return an empty response list with a TCG status code of RESPONSE_OVERFLOW in that method's response status list.

3.3.2 Supported Security Protocols

The TPer SHALL support:

- IF-RECV commands with a Security Protocol values of 0x00, 0x01, 0x02.
- IF-SEND commands with a Security Protocol values of 0x01, 0x02.

3.3.3 ComIDs

For the purpose of communication using Security Protocol 0x01, the TPer SHALL:

- support at least one statically allocated ComID for Synchronous Protocol communication.

- have the ComID Extension values = 0x0000 for all statically allocated ComIDs.
- keep all statically allocated ComIDs in the Active state.

When the TPer receives an IF-SEND or IF-RECV with an inactive or unsupported ComID, the TPer SHALL either:

- terminate the command as defined in [4] with "Other Invalid Command Parameter", or
- follow the requirements defined in [2] for "IF-SEND to Inactive or Unsupported Reserved ComID" or "IF-RECV to Inactive or Unsupported Reserved ComID".

ComIDs SHALL be assigned based on the allocation presented in Table 15.

Table 15 ComID Assignments

ComID	Description
0x0000	Reserved
0x0001	Level 0 Device Discovery
0x0002-0x0003	Reserved for TCG
0x0004	TPER_RESET command
0x0005-0x07FF	Reserved for TCG
0x0800-0x0FFF	Vendor Unique
0x1000-0xFFFF	ComID management (Protocol ID=0x01 and 0x02)

3.3.4 Synchronous Protocol

The TPer SHALL support the Synchronous Protocol. Refer to [2] for details.

3.3.4.1 Payload Encoding

3.3.4.1.1 Stream Encoding Modifications

The TPer SHALL support tokens listed in Table 16. If an unsupported token is encountered, the TPer SHALL treat the token as a streaming protocol violation and return an error per the definition in section 3.3.4.1.3.

Table 16 Supported Tokens

Token	Acronym
Tiny atom	N/A
Short atom	N/A
Medium atom	N/A
Long atom	N/A
Start List	SL
End List	EL
Start Name	SN
End Name	EN
Call	CALL
End of Data	EOD

End of session	EOS
Start transaction	ST
End of transaction	ET
Empty atom	MT

The TPer SHALL support the above token atoms with the B bit set to zero or one and the S bit set to zero.

3.3.4.1.2 TCG Packets

Within a single IF-SEND/IF-RECV command, the TPer SHALL support a ComPacket containing one Packet, which contains one Subpacket. The host may discover TPer support of capabilities beyond this requirement in the parameters returned in response to a `Properties` method.

The TPer MAY ignore Credit Control Subpackets sent by the host. The host may discover TPer support of Credit Management with Level 0 Discovery. For more details refer to Section 3.1.1 Level 0 Discovery (M)

The TPer MAY ignore the AckType and Acknowledgement fields in the Packet header on commands from the host and set these fields to zero in its responses to the host. The host may discover TPer support of the TCG packet acknowledgement/retry mechanism with Level 0 Discovery. For more details refer to Section 3.1.1 Level 0 Discovery (M)

The TPer MAY ignore packet sequence numbering and not enforce any sequencing behavior. Refer to [2] for details on discovery of packet sequence numbering support.

3.3.4.1.3 Payload Error Response

The TPer SHALL respond according to the following rules if it encounters a streaming protocol violation:

- If the error is on Session Manager or is such that the TPer cannot resolve a valid session ID from the payload (i.e. errors in the ComPacket header or Packet header), then the TPer SHALL discard the payload and immediately transition to the “Awaiting IF-SEND” state.
- If the error occurs after the TPer has resolved the session ID, then the TPer SHALL abort the session and MAY prepare a `CloseSession` method for retrieval by the host.

3.3.5 Storage Device Resets

3.3.5.1 Interface Resets

Interface resets that generate TCG reset events are defined in [4].

Interface initiated TCG reset events SHALL result in:

1. All open sessions SHALL be aborted;
2. All uncommitted transactions SHALL be aborted;
3. All pending session startup activities SHALL be aborted;
4. All TCG command and response buffers SHALL be invalidated;
5. All related method processing SHALL be aborted;
6. For each ComID, the state of the synchronous protocol stack SHALL transition to “Awaiting IF-SEND” state;
7. No notification of these events SHALL be sent to the host.

3.3.5.2 TCG Reset Events

Table 17 replaces the definition of TCG reset_types that are defined in [2]:

Table 17 reset_types

Enumeration value	Associated Value
0	Power Cycle
1	Hardware
2	HotPlug
3	Programmatic
4-15	Reserved
16-31	Vendor Unique

3.3.6 Protocol Stack Reset Commands (M)

An IF-SEND containing a Protocol Stack Reset Command SHALL be supported.

Refer to [2] for details.

4 Opal SSC-compliant Functions and SPs

4.1 Session Manager

4.1.1 Methods

4.1.1.1 Properties (M)

An Opal compliant Storage Device SHALL support the `Properties` method. The requirements for support of the various TPer and Host properties, and the requirements for their values, are shown in Table 18.

Table 18 Properties Requirements

Property Name	TPer Property Requirements and Values Reported	Host Property Requirements and Values Accepted
MaxComPacketSize	(M) 2048 minimum	(M) Initial Assumption: 2048 Minimum allowed: 2048 Maximum allowed: VU
MaxResponseComPacketSize	(M) 2048 minimum	(N) Although this is a legal host property, there is no requirement for the TPer to use it. The TPer MAY ignore this host property and not list it in the HostProperties result of the `Properties` method response.
MaxPacketSize	(M) 2028 minimum	(M) Initial Assumption: 2028 Minimum allowed: 2028 Maximum allowed: VU
MaxIndTokenSize	(M) 1992 minimum	(M) Initial Assumption: 1992 Minimum allowed: 1992 Maximum allowed: VU
MaxPackets	(M) 1 minimum	(M) Initial Assumption: 1 Minimum allowed: 1 Maximum allowed: VU
MaxSubpackets	(M) 1 minimum	(M)

		Initial Assumption: 1 Minimum allowed: 1 Maximum allowed: VU
MaxMethods	(M) 1 minimum	(M) Initial Assumption: 1 Minimum allowed: 1 Maximum allowed: VU
MaxSessions	(M) 1 minimum	N/A – not a host property
MaxAuthentications	(M) 2 minimum	N/A – not a host property
MaxTransactionLimit	(M) 1 minimum	N/A – not a host property
DefSessionTimeout	(M) VU	N/A – not a host property

4.1.1.2 StartSession (M)

An Opal-compliant SD SHALL support the following parameters for the `StartSession` method:

- HostSessionID
- SPID
- Write
- HostChallenge
- HostSigningAuthority

For an Opal-compliant SD, a value of “True” for the Write parameter SHALL be supported.

For an Opal-compliant SD, a value of “False” (i.e. read only session) for the Write parameter may or may not be supported.

4.1.1.3 SyncSession (M)

An Opal-compliant SD SHALL support the following parameters for the `SyncSession` method:

- HostSessionID
- SPSessionID

4.1.1.4 CloseSession (O)

An Opal-Compliant SD MAY support the `CloseSession` method.

4.2 Admin SP

The Admin SP includes the Base Template and the Admin Template.

4.2.1 Base Template Tables

All tables included in the following subsections are Mandatory.

4.2.1.1 SPInfo (M)

The SPInfo Table is defined in [2], and Table 19 defines the Preconfiguration Data for the SPInfo Table.

Table 19 Admin SP - SPInfo Table Preconfiguration

UID	SPID	Name	Size	SizeInUse	SPSessionTimeout	Enabled
00 00 00 02 00 00 00 01	00 00 02 05 00 00 00 01	"Admin"				T

4.2.1.2 SPTemplates (M)

The SPTemplates Table is defined in [2], and Table 20 defines the Preconfiguration Data for the SPTemplates Table.

*ST1 means this version number or any version number that complies with this SSC.

Table 20 Admin SP - SPTemplates Table Preconfiguration

UID	TemplateID	Name	Version
00 00 00 03 00 00 00 01	00 00 02 04 00 00 00 01	"Base"	00 00 00 02 *ST1
00 00 00 03 00 00 00 02	00 00 02 04 00 00 00 02	"Admin"	00 00 00 02 *ST1

4.2.1.3 Table (M)

The Table Table is defined in [2], and Table 21 defines the Preconfiguration Data for the Table Table.

Refer to section 5.3 for a description and requirements of the MandatoryWriteGranularity and RecommendedAccessGranularity columns.

Table 21 Admin SP - Table Table Preconfiguration

UID	Name	CommonName	TemplateID	Kind	Column	NumColumns	Rows	RowsFree	RowBytes	LastID	MinSize	MaxSize	MandatoryWrite Granularity	RecommendedAccess
00 00 00 01 00 00 00 01	"Table"			Object									0	0

UID	Name	CommonName	TemplateID	Kind	Column	NumColumns	Rows	RowsFree	RowBytes	LastID	MinSize	MaxSize	MandatoryWrite Granularity	RecommendedAccess
00 00 00 01 00 00 00 02	“SPInfo”			Object									0	0
00 00 00 01 00 00 00 03	“SPTemplates”			Object									0	0
00 00 00 01 00 00 00 06	"MethodID"			Object									0	0
00 00 00 01 00 00 00 07	"AccessControl"			Object									0	0
00 00 00 01 00 00 00 08	"ACE"			Object									0	0
00 00 00 01 00 00 00 09	"Authority"			Object									0	0
00 00 00 01 00 00 00 0B	"C_PIN"			Object									0	0
00 00 00 01 00 00 02 01	"TPerInfo"			Object									0	0
00 00 00 01 00 00 02 04	"Template"			Object									0	0
00 00 00 01 00 00 02 05	"SP"			Object									0	0
00 00 00 01 00 00 11 01	“DataRemovalMechanism”			Object									0	0

Start of Informative Comment

[2] states, “The Table table in the Admin SP includes a row for each table that the TPer supports, in addition to a row for each table that exists in the Admin SP.” However, the Opal SSC requires only the tables from the Admin SP to be included in the Admin SP’s Table table, as indicated in Table 21.

End of Informative Comment

4.2.1.4 MethodID (M)

The MethodID Table is defined in [2], and Table 22 defines the Preconfiguration Data for the MethodID Table.

*MT1: refer to section 5.1.2 for details on the requirements for supporting Revert.

*MT2: refer to section 5.1.1 for details on the requirements for supporting Activate.

.

Table 22 Admin SP - MethodID Table Preconfiguration

UID	Name	CommonName	TemplateID
00 00 00 06 00 00 00 08	"Next"		
00 00 00 06 00 00 00 0D	"GetACL"		
00 00 00 06 00 00 00 16	"Get"		
00 00 00 06 00 00 00 17	"Set"		
00 00 00 06 00 00 00 1C	"Authenticate"		
00 00 00 06 00 00 02 02 *MT1	"Revert"		
00 00 00 06 00 00 02 03 *MT2	"Activate"		
00 00 00 06 00 00 06 01	“Random”		

4.2.1.5 AccessControl (M)

Table 23 contains Optional rows identified by (O).

Notation:

*AC1: the notation of “TT TT TT TT” represents a shorthand for the LSBs of the Table object UIDs

*AC2: the notation of “TT TT TT TT” represents a shorthand for the LSBs of the SPTemplates object UIDs

*AC3: the notation of “TT TT TT TT” represents a shorthand for the LSBs of the MethodID object UIDs

*AC4: the notation of “TT TT TT TT” represents a shorthand for the LSBs of the ACE object UIDs

*AC5: the notation of “TT TT TT TT” represents a shorthand for the LSBs of the Authority object UIDs

*AC6: the notation of “TT TT TT TT” represents a shorthand for the LSBs of the Template object UIDs

*AC7: the notation of “TT TT TT TT” represents a shorthand for the LSBs of the SP object UIDs

Start of Informative Comment

*AC8: refer to section 5.1.2 for details on the requirements for supporting Revert

*AC9: refer to section 5.1.1 for details on the requirements for supporting Activate

End of Informative Comment

The `InvokingID`, `MethodID` and `GetACLACL` columns are a special case. Although they are marked as Read-Only with fixed access control, the access control for invocation of the `Get` method is (N).

The `ACL` column is readable only via the `GetACL` method.

Table 23 Admin SP - AccessControl Table Preconfiguration

Table association - Informative text	**UID**	**InvokingID**	InvokingID Name - informative text	**MethodID**	**CommonName**	**ACL**	**Log**	**AddACEACL**	**RemoveACEACL**	**GetACLACL**	**DeleteMethodACL**	**AddACELog**	**RemoveACELog**	**GetACLLog**	**DeleteMethodLog**	**LogTo**
Table																
		00 00 00 01 00 00 00 00	Table	Next		ACE_Anybody				ACE_Anybody						
		00 00 00 01 TT TT TT TT *AC1	TableObj	Get		ACE_Anybody				ACE_Anybody						
SPInfo																
		00 00 00 02 00 00 00 01	SPInfoObj	Get		ACE_Anybody				ACE_Anybody						
SPTemplates																

Table association - Informative text	UID	InvokingID	InvokingID Name - informative text	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
		00 00 00 03 00 00 00 00	SPTemplates	Next		ACE_Anybody				ACE_Anybody						
		00 00 00 03 TT TT TT TT *AC2	SPTemplatesObj	Get		ACE_Anybody				ACE_Anybody						
MethodID																
		00 00 00 06 00 00 00 00	MethodID	Next		ACE_Anybody				ACE_Anybody						
		00 00 00 06 TT TT TT TT *AC3	MethodIDObj	Get		ACE_Anybody				ACE_Anybody						
ACE																
		00 00 00 08 00 00 00 00	ACE	Next		ACE_Anybody				ACE_Anybody						
		00 00 00 08 TT TT TT TT *AC4	ACEObj	Get		ACE_Anybody				ACE_Anybody						
Authority																
		00 00 00 09 00 00 00 00	Authority	Next		ACE_Anybody				ACE_Anybody						

Table association - Informative text	UID	InvokingID	InvokingID Name - informative text	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
		00 00 00 09 TT TT TT TT *AC5	AuthorityObj	Get		ACE_Anybody				ACE_Anybody						
		00 00 00 09 00 00 00 03	Makers	Set		ACE_Set_Enabled				ACE_Anybody						
		00 00 00 09 00 00 02 01	Admin1	Set		ACE_Set_Enabled				ACE_Anybody						
		00 00 00 09 00 00 02 00 (+XX)	AdminXX	Set		ACE_Set_Enabled				ACE_Anybody						

Table association - Informative text	UID	InvokingID	InvokingID Name - informative text	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
C_PIN																
		00 00 00 0B 00 00 00 00	C_PIN	Next		ACE_Anybody				ACE_Anybody						
		00 00 00 0B 00 00 00 01	C_PIN_SID	Get		ACE_C_PIN_SID_Get_NOPIN				ACE_Anybody						
		00 00 00 0B 00 00 00 01	C_PIN_SID	Set		ACE_C_PIN_SID_Set_PIN				ACE_Anybody						
		00 00 00 0B 00 00 84 02	C_PIN_MSID	Get		ACE_C_PIN_MSID_Get_PIN				ACE_Anybody						
		00 00 00 0B 00 00 02 01	C_PIN_Admin1	Get		ACE_C_PIN_SID_Get_NOPIN				ACE_Anybody						

Table association - Informative text	UID	InvokingID	InvokingID Name - informative text	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
		00 00 00 0B 00 00 02 00 (+XX)	C_PIN_AdminXX	Get		ACE_C_PIN_SID_Get_NOPIN				ACE_Anybody						
		00 00 00 0B 00 00 02 01	C_PIN_Admin1	Set		ACE_C_PIN_Admins_Set_PIN				ACE_Anybody						
		00 00 00 0B 00 00 02 00 (+XX)	C_PIN_AdminXX	Set		ACE_C_PIN_Admins_Set_PIN				ACE_Anybody						
TPerInfo																
		00 00 02 01 00 03 00 01	TPerInfoObj	Get		ACE_Anybody				ACE_Anybody						

Table association - Informative text	UID	InvokingID	InvokingID Name - informative text	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
		00 00 02 01 00 03 00 01	TPerInfoObj	Set		ACE_TPerInfo_Set_ProgrammaticResetEnabl e				ACE_Anybody						
Template																
		00 00 02 04 00 00 00 00	Template	Next		ACE_Anybody				ACE_Anybody						
		00 00 02 04 TT TT TT TT *AC6	TemplateObj	Get		ACE_Anybody				ACE_Anybody						
SP																
		00 00 00 00 00 00 00 01	ThisSP	Authenticate		ACE_Anybody				ACE_Anybody						
		00 00 00 00 00 00 00 01	ThisSP	Random		ACE_Anybody				ACE_Anybody						
		00 00 02 05 00 00 00 00	SP	Next		ACE_Anybody				ACE_Anybody						

Table association - Informative text	UID	InvokingID	InvokingID Name - informative text	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
		00 00 02 05 TT TT TT TT *AC7	SPObj	Get		ACE_Anybody				ACE_Anybody						
*AC8		00 00 02 05 TT TT TT TT *AC7	SPObj	Revert		ACE_SP_SID, ACE_Admin				ACE_Anybody						
*AC9		00 00 02 05 TT TT TT TT *AC7	SPObj	Activate		ACE_SP_SID				ACE_Anybody						
DataRemovalMechanism																
		00 00 11 01 00 00 00 01	DataRemovalMechanismObj	Get		ACE_Anybody				ACE_Anybody						
		00 00 11 01 00 00 00 01	DataRemovalMechanismObj	Set		ACE_DataRemovalMechanism_Set				ACE_Anybody						

4.2.1.6 ACE (M)

Table 24 contains Optional rows designated with (O).

Start of Informative Comment

*ACE1 means that row is (M) if the TPer supports either Activate or Revert, and (N) otherwise.

End of Informative Comment

Table 24 Admin SP - ACE Table Preconfiguration

Table Association - Informative text	UID	Name	CommonName	BooleanExpr	Columns
BaseACEs					
	00 00 00 08 00 00 00 01	"ACE_Anybody"		Anybody	All
	00 00 00 08 00 00 00 02	"ACE_Admin"		Admins	All
Authority					
	00 00 00 08 00 03 00 01	"ACE_Set_Enabled"		SID	Enabled
C_PIN					
	00 00 00 08 00 00 8C 02	"ACE_C_PIN_SID_Get_NOPIN"		Admins OR SID	UID, CharSet, TryLimit, Tries, Persistence
	00 00 00 08 00 00 8C 03	"ACE_C_PIN_SID_Set_PIN"		SID	PIN
	00 00 00 08 00 00 8C 04	"ACE_C_PIN_MSID_Get_PIN"		Anybody	UID, PIN
	00 00 00 08 00 03 A0 01	"ACE_C_PIN_Admins_Set_PIN"		Admins OR SID	PIN
TPerInfo					
	00 00 00 08 00 03 00 03	"ACE_TPerInfo_Set_ProgrammaticResetEnable"		SID	ProgrammaticResetEnable
SP					
*ACE1	00 00 00 08 00 03 00 02	"ACE_SP_SID"		SID	All
DataRemovalMechanism					
*ACE1	00 00 00 08 00 05 00 01	"ACE_DataRemovalMechanism_Set_ActiveDataRemovalMechanism"		Admins OR SID	ActiveDataRemoval Mechanism

4.2.1.7 Authority (M)

The Authority Table is defined in [2], and Table 25 defines the Preconfiguration Data for the Authority Table.

Note:

- Admin1 (M) is required; any additional Admin authorities are (O)

Table 25 Admin SP - Authority Table Preconfiguration

UID	Name	CommonName	IsClass	Class	Enabled	Secure	HashAndSign	PresentCertificat	Operation	Credential	ResponseSign	ResponseExch	ClockStart	ClockEnd	Limit	Uses	Log	LogTo
00 00 00 09 00 00 00 01	"Anybody"		F	Null	T	None	None	F	None	Null	Null	Null						
00 00 00 09 00 00 00 02	"Admins"		T	Null	T	None	None	F	None	Null	Null	Null						
00 00 00 09 00 00 00 03	"Makers"		T	Null	T	None	None	F	None	Null	Null	Null						
00 00 00 09 00 00 00 06	"SID"		F	Null	T	None	None	F	Password	C_PIN_SID	Null	Null						
00 00 00 09 00 00 02 01	"Admin1"		F	Admins	F	None	None	F	Password	C_PIN_Admin1	Null	Null						
00 00 00 09 00 00 02 00 (+XX)[1] (O)	"AdminXX"		F	Admins	F	None	None	F	Password	C_PIN_AdminXX	Null	Null						

4.2.1.8 C_PIN (M)

The C_PIN Table is defined in [2], and Table 26 defines the Preconfiguration Data for the C_PIN Table.

Table 26 Admin SP - C_PIN Table Preconfiguration

UID	Name	CommonName	PIN	CharSet	TryLimit	Tries	Persistence
00 00 00 0B 00 00 00 01	"C_PIN_SID"		VU	Null	**VU**	**VU**	FALSE

UID	Name	CommonName	PIN	CharSet	TryLimit	Tries	Persistence
00 00 00 0B 00 00 84 02	"C_PIN_MSID"		**MSID**				
00 00 00 0B 00 00 02 01	"C_PIN_Admin1"		“”	Null	**0**	**0**	FALSE
00 00 00 0B 00 00 02 00 (+XX) (O)	"C_PIN_AdminXX"		“”	Null	**0**	**0**	FALSE

For SDs that will be used in environments where an automated take ownership process is required, the initial PIN column value of C_PIN_SID SHALL be set to the PIN column value of C_PIN_MSID. In order to allow for alternative take ownership processes, the initial PIN column value of C_PIN_SID MAY be Vendor Unique (VU).

Start of Informative Comment

Several activation / take ownership models are possible. The simplest model, which is the only model supported by Opal v1.00, is a process whereby the host discovers the initial C_PIN_SID PIN value by performing a Get operation on the C_PIN_MSID object. This model requires that the initial C_PIN_SID PIN be the value of the C_PIN_MSID PIN.

Opal v2.00 allows the initial C_PIN_SID PIN value to be vendor unique in order to allow for alternative activation / take ownership models. Such models require that the C_PIN_SID PIN be retrieved in a way that is beyond the scope of this specification.

Before a device vendor chooses to implement an activation / take ownership model based on a vendor unique SID PIN, the SD vendor must undertake due diligence to ensure that the ecosystem exists to support such an activation / take ownership model. Having a C_PIN_SID PIN value that is different from the C_PIN_MSID PIN value may have serious ramifications, such as the inability to take ownership of the SD.

See section 5.1.2.2.1 for an explanation of how Revert affects the value of the C_PIN_SID PIN column.

End of Informative Comment

4.2.2 Base Template Methods

Refer to section 4.2.1.4 for supported methods.

4.2.3 Admin Template Tables

4.2.3.1 TPerInfo (M)

The TPerInfo table has the column defined in Table 27, in addition to those defined in [2]:

Table 27 Admin SP – TPerInfo Columns

Column Number	Column Name	IsUnique	Colum Type
0x08	ProgrammaticResetEnable		boolean

- **ProgrammaticResetEnable**
  This column indicates whether support for programmatic resets is enabled or not. If ProgrammaticResetEnable is TRUE, then the TPER_RESET command is enabled. If ProgrammaticResetEnable is FALSE, then the TPER_RESET command is not enabled. This column is readable by Anybody and modifiable by the SID authority.

*TP1 means that the value in the GUDID column SHALL comply with the format defined in [2].

*TP2 means that this version or any version that supports the defined features in this SSC.

*TP3 means that the SSC column is a list of names and SHALL have “Opal” as one of the list elements.

Table 28 Admin SP - TPerInfo Table Preconfiguration

UID	Bytes	GUDID	Generation	Firmware Version	ProtocolVersion	SpaceForIssuance	SSC	ProgrammaticResetEnable
00 00 02 01 00 03 00 01		**VU** *TP1			1 *TP2		[“Opal”] *TP3	FALSE

4.2.3.2 Template (M)

The Template Table is defined in [2], and Table 29 defines the Preconfiguration Data for the Template Table.

Table 29 Admin SP - Template Table Preconfiguration

UID	Name	Revision Number	Instances	MaxInstances
00 00 02 04 00 00 00 01	"Base"	1	**VU**	**VU**
00 00 02 04 00 00 00 02	"Admin"	1	1	1
00 00 02 04 00 00 00 06	"Locking"	1	1	1

4.2.3.3 SP (M)

The SP Table is defined in [2], and Table 30 defines the Preconfiguration Data for the SP Table.

*SP1 means that this row only exists in the Admin SP's OFS when the Locking SP is created by the manufacturer.

Table 30 Admin SP - SP Table Preconfiguration

UID	Name	ORG	EffectiveAuth	DateOfIssue	Bytes	LifeCycle	Frozen
00 00 02 05 00 00 00 01	"Admin"					Manufactured	FALSE
00 00 02 05 00 00 00 02 *SP1	"Locking"					Manufactured-Inactive	FALSE

4.2.4 Admin Template Methods

Refer to section 4.2.1.4 for supported methods.

4.2.5 Opal Additional Column Types

4.2.5.1 Data_removal_mechanism

The `data_removal_mechanism` type is defined in Table 31 for Opal:

Table 31 data_removal_mechanism Type Table Addition

UID	Name	Format
00 00 00 05 00 00 04 20	data_removal_mechanism	Enumeration_Type, 0, 7

Table 32 defines the enumeration values. The mechanisms associated with each Enumeration Value are defined in Table 11.

Table 32 data_removal_mechanism Enumeration Values

Enumeration Value	Associated Value
0	Overwrite Data Erase
1	Block Erase
2	Cryptographic Erase
3 – 4	Reserved
5	Vendor Specific Erase
6-7	Reserved

4.2.6 Opal Additional Data Structures

4.2.6.1 DataRemovalMechanism (ObjectTable)

The `DataRemovalMchanism` table is defined in Table 33

Table 33 DataRemovalMechansim Table Description

Column Number	**Column Name**	**IsUnique**	**Column Type**
0x00	UID		uid
0x01	ActiveDataRemovalMechanism		data_removal_mechanism

4.2.6.1.1 UID

This is the unique identifier of this row in the `DataRemovalMechanism` table.

This column SHALL NOT be modifiable by the host.

4.2.6.1.2 ActiveDataRemovalMechanism

This column value selects which Data Removal Mechanism in the Supported Data Removal Mechanism field in the Supported Data Removal Mechanism feature descriptor is active and will be used to remove data upon execution of the `Revert` method or the `RevertSP` method or the `GenKey` method. If an attempt is made to set the

ActiveDataRemovalMechanism column value to an unsupported value of the data_removal_mechanism type, then the `Set` method invocation SHALL result in the method failing with the status INVALID_PARAMETER.

4.2.7 Opal Additional Tables

4.2.7.1 DataRemovalMechansim (M)

The `DataRemovalMechanism` table SHALL contain exactly one row with `UID=0x00 0x00 0x11 0x01 0x00 0x00 0x00 0x01`. The `DataRemovalMechanism` table SHALL be supported (see Table 34).

Table 34 Admin SP – DataRemovalMechansim Table Preconfiguration

UID	ActiveDataRemovalMechanism
00 00 11 01 00 00 00 01	VU

4.2.8 Crypto Template Tables

An Opal SSC compliant Storage Device is not required to support any Crypto template tables.

4.2.9 Crypto Template Methods

Refer to section 4.2.1.4 for supported methods.

4.2.9.1 Random

The TPer SHALL implement the `Random` method with the constraints stated in this subsection. TPer support of the following parameters is Mandatory:

- `Count`

Attempts to use unsupported parameters SHALL result in a method failure response with TCG status INVALID_PARAMETER. The TPer SHALL support `Count` parameter values less than or equal to 32.

4.3 Locking SP

4.3.1 Base Template Tables

All tables defined with (M) in section titles are Mandatory.

4.3.1.1 SPInfo (M)

The SPInfo Table is defined in [2], and Table 35 defines the Preconfiguration Data for the SPInfo Table.

Table 35 Locking SP - SPInfo Table Preconfiguration

UID	SPID	Name	Size	SizeInUse	SPSessionTimeout	Enabled
00 00 00 02 00 00 00 01	00 00 02 05 00 00 00 02	"Locking"				T

4.3.1.2 SPTemplates (M)

The SPTemplates Table is defined in [2], and Table 36 defines the Preconfiguration Data for the SPTemplates Table.

*SP1 means that this version number or any number that supports the defined features in this SSC

Table 36 Locking SP - SPTemplates Table Preconfiguration

UID	TemplateID	Name	Version
00 00 00 03 00 00 00 01	00 00 02 04 00 00 00 01	"Base"	00 00 00 02 *SP1
00 00 00 03 00 00 00 02	00 00 02 04 00 00 00 06	"Locking"	00 00 00 02 *SP1

4.3.1.3 Table (M)

The Table Table is defined in [2], and Table 37 defines the Preconfiguration Data for the Table Table.

Table 37 contains Optional rows designated with (O).

TT1 means that only one of the two K_AES tables is required

Refer to section 5.3 for a description and requirements of the MandatoryWriteGranularity and RecommendedAccessGranularity columns.

Table 37 Locking SP - Table Table Preconfiguration

UID	Name	CommonName	TemplateID	Kind	Column	NumColumns	Rows	RowsFree	RowBytes	LastID	MinSize	MaxSize	MandatoryWrite	RecommendedAccess
00 00 00 01 00 00 00 01	"Table"			Object									0	0
00 00 00 01 00 00 00 02	"SPInfo"			Object									0	0
00 00 00 01 00 00 00 03	"SPTemplates"			Object									0	0
00 00 00 01 00 00 00 06	"MethodID"			Object									0	0
00 00 00 01 00 00 00 07	"AccessControl"			Object									0	0
00 00 00 01 00 00 00 08	"ACE"			Object									0	0
00 00 00 01 00 00 00 09	"Authority"			Object									0	0
00 00 00 01 00 00 00 0B	"C_PIN"			Object									0	0

UID	Name	CommonName	TemplateID	Kind	Column	NumColumns	Rows	RowsFree	RowBytes	LastID	MinSize	MaxSize	MandatoryWrite	RecommendedAccess
00 00 00 01 00 00 00 1D	"SecretProtect"			Object									0	0
00 00 00 01 00 00 08 01	"LockingInfo"			Object									0	0
00 00 00 01 00 00 08 02	"Locking"			Object									0	0
00 00 00 01 00 00 08 03	"MBRControl"			Object									0	0
00 00 00 01 00 00 08 04	"MBR"			Byte			0x08000000 min						VU	VU
00 00 00 01 00 00 08 05 *TT1	"K_AES_128"			Object									0	0
00 00 00 01 00 00 08 06 *TT1	"K_AES_256"			Object									0	0
00 00 00 01 00 00 10 01	"DataStore"			Byte			0x00A00000 min						VU	VU

4.3.1.4 Type (N)

The `Type` table is not required (N) by Opal. The following types as defined by [2] SHALL meet the following requirements:

- The "boolean_ACE" type (00000005 0000040E) SHALL include the OR Boolean operator.
- The "AC_element" type (00000005 00000801) SHALL support at least 23 entries (8 User authorities, 4 Admin authorities, and 11 Boolean operators).

4.3.1.5 MethodID (M)

The MethodID Table is defined in [2], and Table 38 defines the Preconfiguration Data for the MethodID Table.

*MT1 means refer to section 5.1.2.3 for details on the requirements for supporting RevertSP.

Table 38 Locking SP - MethodID Table Preconfiguration

UID	Name	CommonName	TemplateID
00 00 00 06 00 00 00 08	"Next"		

UID	Name	CommonName	TemplateID
00 00 00 06 00 00 00 0D	"GetACL"		
00 00 00 06 00 00 00 10	"GenKey"		
00 00 00 06 00 00 00 11 *MT1	"RevertSP"		
00 00 00 06 00 00 00 16	"Get"		
00 00 00 06 00 00 00 17	"Set"		
00 00 00 06 00 00 00 1C	"Authenticate"		
00 00 00 06 00 00 06 01	“Random”		

4.3.1.6 AccessControl (M)

Table 39 contains Optional rows designated with (O).

Start of Informative Comment

*AC1: refer to section 5.1.2.3 for details on the requirements for supporting RevertSP

*AC8: the notation of “TT TT TT TT” represents a shorthand for the LSBs of the SecretProtect object UIDs

End of Informative Comment

*AC2: the notation of “TT TT TT TT” represents a shorthand for the LSBs of the Table object UIDs

*AC3: the notation of “TT TT TT TT” represents a shorthand for the LSBs of the SPTemplates object UIDs

*AC4: the notation of “TT TT TT TT” represents a shorthand for the LSBs of the MethodID object UIDs

*AC5: the notation of “TT TT TT TT” represents a shorthand for the LSBs of the ACE object UIDs

*AC6: only K_AES_128 or K_AES_256 related rows are Mandatory

*AC7: the notation of “TT TT TT TT” represents a shorthand for the LSB of the Authority object UIDs

Notes:

- The `AccessControl` Table is different from any other table defined in this specification. Although cells in this table are marked as Read-Only with fixed access control, the access control for invocation of the `Get` method is (N).
- The `ACL` column is readable only via the `GetACL` method.

Table 39 Locking SP - AccessControl Table Preconfiguration

Table Association - informative only	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
SP																
		00 00 00 00 00 00 00 01	ThisSP	Authenticate		ACE_Anybody				ACE_Anybody						
		00 00 00 00 00 00 00 01	ThisSP	Random		ACE_Anybody				ACE_Anybody						
*AC1		00 00 00 00 00 00 00 01	ThisSP	RevertSP		ACE_Admin				ACE_Anybody						
Table																
		00 00 00 01 00 00 00 00	Table	Next		ACE_Anybody				ACE_Anybody						
		00 00 00 01 TT TT TT TT *AC2	TableObj	Get		ACE_Anybody				ACE_Anybody						
SPInfo																
		00 00 00 02 00 00 00 01	SPInfoObj	Get		ACE_Anybody				ACE_Anybody						
SPTemplates																
		00 00 00 03 00 00 00 00	SPTemplates	Next		ACE_Anybody				ACE_Anybody						

Table Association - informative only	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
		00 00 00 03 TT TT TT TT *AC3	SPTemplatesObj	Get		ACE_Anybody				ACE_Anybody						
MethodID																
		00 00 00 06 00 00 00 00	MethodID	Next		ACE_Anybody				ACE_Anybody						
		00 00 00 06 TT TT TT TT *AC4	MethodIDObj	Get		ACE_Anybody				ACE_Anybody						
ACE																
		00 00 00 08 00 00 00 00	ACE	Next		ACE_Anybody				ACE_Anybody						
		00 00 00 08 TT TT TT TT *AC5	ACEObj	Get		ACE_ACE_Get_All				ACE_Anybody						
		00 00 00 08 00 03 80 00	ACE_ACE_Get_All	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						

Table Association - informative only	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
		00 00 00 08 00 03 90 00	ACE_Authority_Get_All	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						
		00 00 00 08 00 03 A8 01	ACE_C_PIN_User1_Set_PIN	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						
		00 00 00 08 00 03 A8 00 (+MMMM)	ACE_C_PIN_UserMMMM_Set_PIN	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						
		00 00 00 08 00 04 40 01	ACE_User1_Set_CommonName	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						

Table Association - informative only	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
		00 00 00 08 00 04 40 00 (+MMMM)	ACE_UserMMMM_Set_CommonName	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						
*AC6		00 00 00 08 00 03 B0 00	ACE_K_AES_128_GlobalRange_GenKey	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						
*AC6		00 00 00 08 00 03 B0 01	ACE_K_AES_128_Range1_GenKey	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						

Table Association - informative only	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
*AC6		00 00 00 08 00 03 B0 00 (+NNNN)	ACE_K_AES_128_RangeNNNN_GenKey	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						
*AC6		00 00 00 08 00 03 B8 00	ACE_K_AES_256_GlobalRange_GenKey	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						
*AC6		00 00 00 08 00 03 B8 01	ACE_K_AES_256_Range1_GenKey	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						

Table Association - informative only	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
*AC6		00 00 00 08 00 03 B8 00 (+NNNN)	ACE_K_AES_256_RangeNNNN _GenKey	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						
		00 00 00 08 00 03 D0 00	ACE_Locking_GlobalRange_Get_ RangeStartToActiveKey	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						
		00 00 00 08 00 03 D0 01	ACE_Locking_Range1_Get_ RangeStartToActiveKey	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						

Table Association - informative only	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
		00 00 00 08 00 03 D0 00 (+NNNN)	ACE_Locking_RangeNNNN_Get_RangeStartToActiveKey	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						
		00 00 00 08 00 03 E0 00	ACE_Locking_GlobalRange_Set_RdLocked	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						
		00 00 00 08 00 03 E0 01	ACE_Locking_Range1_Set_RdLocked	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						

Table Association - informative only	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
		00 00 00 08 00 03 E0 00 (+NNNN)	ACE_Locking_RangeNNNN_Set_RdLocked	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						
		00 00 00 08 00 03 E8 00	ACE_Locking_GlobalRange_Set_WrLocked	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						
		00 00 00 08 00 03 E8 01	ACE_Locking_Range1_Set_WrLocked	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						

Table Association - informative only	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
		00 00 00 08 00 03 E8 00 (+NNNN)	ACE_Locking_RangeNNNN_Set_WrLocked	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						
		00 00 00 08 00 03 F8 01	ACE_MBRControl_Set_DoneToDOR	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						
		00 00 00 08 00 03 FC 00	ACE_DataStore_Get_All	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						

Table Association - informative only	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
		00 00 00 08 00 03 FC 01	ACE_DataStore_Set_All	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						
Authority																
		00 00 00 09 00 00 00 00	Authority	Next		ACE_Anybody				ACE_Anybody						
		00 00 00 09 TT TT TT TT *AC7	AuthorityObj	Get		ACE_Authority_Get_All, ACE_Anybody_Get_CommonName				ACE_Anybody						
		00 00 00 09 00 01 00 01	Admin1	Set		ACE_Admins_Set_CommonName				ACE_Anybody						

Table Association - informative only	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
		00 00 00 09 00 01 00 02	Admin2	Set		ACE_Authority_Set_Enabled, ACE_Admins_Set_CommonName				ACE_Anybody						
		00 00 00 09 00 01 00 00 (+XX XX)	AdminXXXX	Set		ACE_Authority_Set_Enabled, ACE_Admins_Set_CommonName				ACE_Anybody						
		00 00 00 09 00 03 00 01	User1	Set		ACE_Authority_Set_Enabled, ACE_User1_Set_CommonName				ACE_Anybody						
		00 00 00 09 00 03 00 00 (+MMMM)	UserMMMM	Set		ACE_Authority_Set_Enabled, ACE_UserMMMM_Set_CommonName				ACE_Anybody						

Table Association - informative only	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
C_PIN																
		00 00 00 0B 00 00 00 00	C_PIN	Next		ACE_Anybody				ACE_Anybody						
		00 00 00 0B 00 01 00 01	C_PIN_Admin1	Get		ACE_C_PIN_Admins_Get_All_NOPIN				ACE_Anybody						
		00 00 00 0B 00 01 00 00 (+ XX XX)	C_PIN_AdminXXXX	Get		ACE_C_PIN_Admins_Get_All_NOPIN				ACE_Anybody						
		00 00 00 0B 00 03 00 01	C_PIN_User1	Get		ACE_C_PIN_Admins_Get_All_NOPIN				ACE_Anybody						

Table Association - informative only	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
		00 00 00 0B 00 03 00 00 (+MM MM)	C_PIN_UserMMMM	Get		ACE_C_PIN_Admins_Get_All_NOPIN				ACE_Anybody						
		00 00 00 0B 00 01 00 01	C_PIN_Admin1	Set		ACE_C_PIN_Admins_Set_PIN				ACE_Anybody						
		00 00 00 0B 00 01 00 00 (+XX XX)	C_PIN_AdminXXXX	Set		ACE_C_PIN_Admins_Set_PIN				ACE_Anybody						
		00 00 00 0B 00 03 00 01	C_PIN_User1	Set		ACE_C_PIN_User1_Set_PIN				ACE_Anybody						

Table Association - informative only	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
		00 00 00 0B 00 03 00 00 (+MM MM)	C_PIN_UserMMMM	Set		ACE_C_PIN_UserMMMM_Set_PIN				ACE_Anybody						

Table Association - informative only	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
SecretProtect																
		00 00 00 1D 00 00 00 00	SecretProtect	Next		ACE_Anybody				ACE_Anybody						
		00 00 00 1D TT TT TT TT *AC8	SecretProtectObj	Get		ACE_Anybody				ACE_Anybody						
LockingInfo																
		00 00 08 01 00 00 00 01	LockingInfoObj	Get		ACE_Anybody				ACE_Anybody						
Locking																
		00 00 08 02 00 00 00 00	Locking	Next		ACE_Anybody				ACE_Anybody						
		00 00 08 02 00 00 00 01	Locking_GlobalRange	Get		ACE_Locking_GlobalRange_Get_ RangeStartToActiveKey, ACE_Anybody_Get_CommonName				ACE_Anybody						

Table Association - informative only	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
		00 00 08 02 00 03 00 01	Locking_Range1	Get		ACE_Locking_Range1_Get_ RangeStartToActiveKey, ACE_Anybody_Get_CommonName				ACE_Anybody						
		00 00 08 02 00 03 00 00 (+NN NN)	Locking_RangeNNNN	Get		ACE_Locking_RangeNNNN_Get_ RangeStartToActiveKey, ACE_Anybody_Get_CommonName				ACE_Anybody						
		00 00 08 02 00 00 00 01	Locking_GlobalRange	Set		ACE_Locking_GlblRng_Admins_Set, ACE_Locking_GlobalRange_Set_RdLocked, ACE_Locking_GlobalRange_Set_WrLocked, ACE_Admins_Set_CommonName				ACE_Anybody						

Table Association - informative only	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
		00 00 08 02 00 03 00 01	Locking_Range1	Set		ACE_Locking_Admins_RangeStartToLOR, ACE_Locking_Range1_Set_RdLocked, ACE_Locking_Range1_Set_WrLocked, ACE_Admins_Set_CommonName				ACE_Anybody						
		00 00 08 02 00 03 00 00 (+NN NN)	Locking_RangeNNNN	Set		ACE_Locking_Admins_RangeStartToLOR, ACE_Locking_RangeNNNN_Set_RdLocked, ACE_Locking_RangeNNNN_Set_WrLocked, ACE_Admins_Set_CommonName				ACE_Anybody						
MBRControl																
		00 00 08 03 00 00 00 01	MBRControlObj	Get		ACE_Anybody				ACE_Anybody						

Table Association - informative only	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
		00 00 08 03 00 00 00 01	MBRControlObj	Set		ACE_MBRControl_Admins_Set, ACE_MBRControl_Set_DoneToDOR				ACE_Anybody						
MBR																
		00 00 08 04 00 00 00 00	MBR	Get		ACE_Anybody				ACE_Anybody						
		00 00 08 04 00 00 00 00	MBR	Set		ACE_Admin				ACE_Anybody						
K_AES_128																
		00 00 08 05 00 00 00 01	K_AES_128_GlobalRange_Key	Get		ACE_K_AES_Mode				ACE_Anybody						

Table Association - informative only	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
		00 00 08 05 00 03 00 01	K_AES_128_Range1_Key	Get		ACE_K_AES_Mode				ACE_Anybody						
		00 00 08 05 00 03 00 00 (+NN NN)	K_AES_128_RangeNNNN_Key	Get		ACE_K_AES_Mode				ACE_Anybody						
		00 00 08 05 00 00 00 01	K_AES_128_GlobalRange_Key	GenKey		ACE_K_AES_128_GlobalRange_GenKey				ACE_Anybody						

Table Association - informative only	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
		00 00 08 05 00 03 00 01	K_AES_128_Range1_Key	GenKey		ACE_K_AES_128_Range1_GenKey				ACE_Anybody						
		00 00 08 05 00 03 00 00 (+NN NN)	K_AES_128_RangeNNNN_Key	GenKey		ACE_K_AES_128_RangeNNNN_GenKey				ACE_Anybody						
K_AES_256																
		00 00 08 06 00 00 00 01	K_AES_256_GlobalRange_Key	Get		ACE_K_AES_Mode				ACE_Anybody						

Table Association - informative only	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
		00 00 08 06 00 03 00 01	K_AES_256_Range1_Key	Get		ACE_K_AES_Mode				ACE_Anybody						
		00 00 08 06 00 03 00 00 (+NN NN)	K_AES_256_RangeNNNN_Key	Get		ACE_K_AES_Mode				ACE_Anybody						
		00 00 08 06 00 00 00 01	K_AES_256_GlobalRange_Key	GenKey		ACE_K_AES_256_GlobalRange_GenKey				ACE_Anybody						

Table Association - informative only	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
		00 00 08 06 00 03 00 01	K_AES_256_Range1_Key	GenKey		ACE_K_AES_256_Range1_GenKey				ACE_Anybody						
		00 00 08 06 00 03 00 00 (+NN NN)	K_AES_256_RangeNNNN_Key	GenKey		ACE_K_AES_256_RangeNNNN_GenKey				ACE_Anybody						
DataStore																
		00 00 10 01 00 00 00 00	DataStore	Get		ACE_DataStore_Get_All				ACE_Anybody						

Table Association - informative only	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
		00 00 10 01 00 00 00 00	DataStore	Set		ACE_DataStore_Set_All				ACE_Anybody						

4.3.1.7 ACE (M)

Table 40 contains Optional rows designated with (O).

*ACE1 means that the TPer SHALL support the values of “Admins” and “Admins OR UserMMMM” in the BooleanExpr column of each ACE_C_PIN_UserMMMM_Set_PIN ACE. The TPer SHALL fail the `Set` method invocation with status INVALID_PARAMETER if the host attempts to set a value not supported by the TPer.

Table 40 Locking SP - ACE Table Preconfiguration

Table Association -Informative Column	UID	Name	CommonName	BooleanExpr	Columns
Base ACEs					
	00 00 00 08 00 00 00 01	"ACE_Anybody"		Anybody	All
	00 00 00 08 00 00 00 02	"ACE_Admin"		Admins	All
	00 00 00 08 00 00 00 03	"ACE_Anybody_Get_CommonName"		Anybody	UID, CommonName
	00 00 00 08 00 00 00 04	"ACE_Admins_Set_CommonName"		Admins	CommonName
ACE					
	00 00 00 08 00 03 80 00	"ACE_ACE_Get_All"		Admins	All
	00 00 00 08 00 03 80 01	"ACE_ACE_Set_BooleanExpression"		Admins	BooleanExpr
Authority					

Table Association -Informative Column	UID	Name	CommonName	BooleanExpr	Columns
	00 00 00 08 00 03 90 00	"ACE_Authority_Get_All"		Admins	All
	00 00 00 08 00 03 90 01	"ACE_Authority_Set_Enabled"		Admins	Enabled
	00 00 00 08 00 04 40 01	"ACE_User1_Set_CommonName"		Admins	CommonName
	00 00 00 08 00 04 40 00 (+NN NN)	"ACE_UserMMMM_Set_CommonName"		Admins	CommonName
C_PIN					
	00 00 00 08 00 03 A0 00	"ACE_C_PIN_Admins_Get_All_NOPIN"		Admins	UID, CharSet, TryLimit, Tries, Persistence
	00 00 00 08 00 03 A0 01	"ACE_C_PIN_Admins_Set_PIN"		Admins	PIN
	00 00 00 08 00 03 A8 01	"ACE_C_PIN_User1_Set_PIN"		Admins OR User1 *ACE1	PIN
(O)	00 00 00 08 00 03 A8 00 (+MMMM)	"ACE_C_PIN_UserMMMM_Set_PIN"		Admins OR UserMMMM *ACE1	PIN
K_AES					
	00 00 00 08 00 03 BF FF	"ACE_K_AES_Mode"		Anybody	Mode
K_AES_128					
	00 00 00 08 00 03 B0 00	"ACE_K_AES_128_GlobalRange_GenKey"		Admins	All
	00 00 00 08 00 03 B0 01	"ACE_K_AES_128_Range1_GenKey"		Admins	All
(O)	00 00 00 08 00 03 B0 00 (+NNNN)	"ACE_K_AES_128_RangeNNNN_GenKey"		Admins	All

Table Association -Informative Column	UID	Name	CommonName	BooleanExpr	Columns
K_AES_256					
	00 00 00 08 00 03 B8 00	"ACE_K_AES_256_GlobalRange_GenKey"		Admins	All
	00 00 00 08 00 03 B8 01	"ACE_K_AES_256_Range1_GenKey"		Admins	All
	00 00 00 08 00 03 B8 00 (+NNNN)	"ACE_K_AES_256_RangeNNNN_GenKey"		Admins	All
Locking					
	00 00 00 08 00 03 D0 00	"ACE_Locking_GlobalRange_Get_RangeStartToActiveKey"		Admins	RangeStart, RangeLength, ReadLockEnabled, WriteLockEnabled, ReadLocked, WriteLocked, LockOnReset, ActiveKey
	00 00 00 08 00 03 D0 01	"ACE_Locking_Range1_Get_RangeStartToActiveKey"		Admins	RangeStart, RangeLength, ReadLockEnabled, WriteLockEnabled, ReadLocked, WriteLocked, LockOnReset, ActiveKey
	00 00 00 08 00 03 D0 00 (+NNNN)	"ACE_Locking_RangeNNNN_Get_RangeStartToActiveKey"		Admins	RangeStart, RangeLength, ReadLockEnabled, WriteLockEnabled, ReadLocked, WriteLocked, LockOnReset, ActiveKey
	00 00 00 08 00 03 E0 00	"ACE_Locking_GlobalRange_Set_RdLocked"		Admins	ReadLocked
	00 00 00 08 00 03 E0 01	"ACE_Locking_Range1_Set_RdLocked"		Admins	ReadLocked

Table Association -Informative Column	UID	Name	CommonName	BooleanExpr	Columns
	00 00 00 08 00 03 E0 00 (+NNNN)	"ACE_Locking_RangeNNNN_Set_RdLocked"		Admins	ReadLocked
	00 00 00 08 00 03 E8 00	"ACE_Locking_GlobalRange_Set_WrLocked"		Admins	WriteLocked
	00 00 00 08 00 03 E8 01	"ACE_Locking_Range1_Set_WrLocked"		Admins	WriteLocked
	00 00 00 08 00 03 E8 00 (+NNNN)	"ACE_Locking_RangeNNNN_Set_WrLocked"		Admins	WriteLocked
	00 00 00 08 00 03 F0 00	"ACE_Locking_GlblRng_Admins_Set"		Admins	ReadLockEnabled, WriteLockEnabled, ReadLocked, WriteLocked, LockOnReset
	00 00 00 08 00 03 F0 01	"ACE_Locking_Admins_RangeStartToLOR"		Admins	RangeStart, RangeLength, ReadLockEnabled, WriteLockEnabled, ReadLocked, WriteLocked, LockOnReset
MBRControl					
	00 00 00 08 00 03 F8 00	"ACE_MBRControl_Admins_Set"		Admins	Enable, Done, DoneOnReset
	00 00 00 08 00 03 F8 01	"ACE_MBRControl_Set_DoneToDOR"		Admins	Done, DoneOnReset
DataStore					
	00 00 00 08 00 03 FC 00	"ACE_DataStore_Get_All"		Admins	All
	00 00 00 08 00 03 FC 01	"ACE_DataStore_Set_All"		Admins	All

4.3.1.8 Authority (M)

Table 41 contains Optional rows designated with (O).
Notes:

1. Admin1 is required; Admin2 to Admin4 are required but disabled in OFS state. Any additional Admin authorities are (O).
2. User1 through User8 SHALL be implemented.

Table 41 Locking SP - Authority Table Preconfiguration

UID	Name	CommonName	IsClass	Class	Enabled	Secure	HashAndSign	PresentCertificate	Operation	Credential	ResponseSign	ResponseExch	ClockStart	ClockEnd	Limit	Uses	Log	LogTo
00 00 00 09 00 00 00 01	"Anybody"	""	F	Null	T	None	None	F	None	Null	Null	Null						
00 00 00 09 00 00 00 02	"Admins"	""	T	Null	T	None	None	F	None	Null	Null	Null						
00 00 00 09 00 01 00 01	"Admin1"	""	F	Admins	T	None	None	F	Password	C_PIN_Admin1	Null	Null						
00 00 00 09 00 01 00 02	"Admin2"	""	F	Admins	F	None	None	F	Password	C_PIN_Admin2	Null	Null						
00 00 00 09 00 01 00 03	"Admin3"	""	F	Admins	F	None	None	F	Password	C_PIN_Admin3	Null	Null						
00 00 00 09 00 01 00 04	"Admin4"	""	F	Admins	F	None	None	F	Password	C_PIN_Admin4	Null	Null						
00 00 00 09 00 01 00 00 (+XX XX)[1] (O)	"AdminXXXX"	""	F	Admins	F													

UID	Name	CommonName	IsClass	Class	Enabled	Secure	HashAndSign	PresentCertificate	Operation	Credential	ResponseSign	ResponseExch	ClockStart	ClockEnd	Limit	Uses	Log	LogTo
00 00 00 09 00 03 00 00	"Users"	""	T	Null	T	None	None	F	None	Null	Null	Null						
00 00 00 09 00 03 00 01	"User1"	"" -	F	Users	F	None	None	F	Password	C_PIN_User1	Null	Null						
00 00 00 09 00 03 00 00 (+MM MM)[2] (O)	"UserMMMM"	"" -	F	Users	F	None	None	F	Password	C_PIN_UserMMMM	Null	Null						

4.3.1.9 C_PIN (M)

Table 42 includes Optional rows designated with (O)

Notes:

1. If the Locking SP's original life cycle state is Manufactured-Inactive, see 5.1.1.2 for the initial value of C_PIN_Admin1.PIN. If the Locking SP's original life cycle state is Manufactured, then the initial value of C_PIN_Admin1.PIN is the same as the Admin SP's C_PIN_MSID.PIN value.

Table 42 Locking SP - C_PIN Table Preconfiguration

UID	Name	CommonName	PIN	CharSet	TryLimit	Tries	Persistence
00 00 00 0B 00 01 00 01	"C_PIN_Admin1"		SID or MSID[1]	Null	0	0	FALSE
00 00 00 0B 00 01 00 02	"C_PIN_Admin2"		""	Null	0	0	FALSE
00 00 00 0B 00 01 00 03	"C_PIN_Admin3"		""	Null	0	0	FALSE
00 00 00 0B 00 01 00 04	"C_PIN_Admin4"		""	Null	0	0	FALSE
00 00 00 0B 00 01 00 00 (+XX XX) (O)	"C_PIN_AdminXXXX"		""	Null	0	0	FALSE

UID	Name	CommonName	PIN	CharSet	TryLimit	Tries	Persistence
00 00 00 0B 00 03 00 01	"C_PIN_User1"		“”	Null	0	0	FALSE
00 00 00 0B 00 03 00 00 (+MM MM) (O)	"C_PIN_UserMMMM"		“”	Null	0	0	FALSE

4.3.1.10 SecretProtect (M)

At least one of the objects shown in Table 433 SHALL be supported

Table 43 Locking SP - SecretProtect Table Preconfiguration

UID	Table	ColumnNumber	ProtectMechanisms
00 00 00 1D 00 00 00 1D	00 00 00 01 00 00 08 05 (K_AES_128)	0x03	VU
00 00 00 1D 00 00 00 1E	00 00 00 01 00 00 08 06 (K_AES_256)	0x03	VU

Note: The “VU” entries in Table 43 indicate that this specification does not require a specific value to be reported in the ProtectMechanisms cell. It is NOT a requirement to report the “Vendor Unique” `protect_types` value (Refer to [2] for details).

4.3.2 Base Template Methods

Refer to section 4.3.1.5 for supported methods.

4.3.3 Crypto Template Tables

An Opal SSC compliant Storage Device is not required to support any Crypto template tables.

4.3.4 Crypto Template Methods

Refer to section 4.3.1.5 for supported methods.

4.3.4.1 Random

Refer to section 4.2.9.1 for additional constraints imposed on the `Random` method.

4.3.5 Locking Template Tables

4.3.5.1 LockingInfo (M)

The `LockingInfo` table has the columns defined in Table 44, in addition to those defined in [2]:

Table 44 Locking SP – LockingInfo Columns

Column Number	Column Name	IsUnique	Column Type
0x07	AlignmentRequired		boolean
0x08	LogicalBlockSize		uinteger_4
0x09	AlignmentGranularity		uinteger_8
0x0A	LowestAlignedLBA		uniteger_8

- **AlignmentRequired**
  This column indicates whether the TPer requires ranges in the `Locking` table to be aligned (see section 4.3.5.2.1). If AlignmentRequired is TRUE, then the TPer requires ranges to be aligned. If AlignmentRequired is FALSE, then the TPer does not require ranges to be aligned.
  This column SHALL NOT be modifiable by the host and MAY be retrieved by Anybody.
- **LogicalBlockSize**
  This column indicates the number of bytes in a logical block.
  This column SHALL NOT be modifiable by the host and MAY be retrieved by Anybody.
- **AlignmentGranularity**
  This column indicates the number of logical blocks in a group, for alignment purposes (see section 5.4).
  This column SHALL NOT be modifiable by the host and MAY be retrieved by Anybody.
- **LowestAlignedLBA**
  This column indicates the lowest logical block address that is located at the beginning of an alignment granularity group (see section 5.4).
  This column SHALL NOT be modifiable by the host and MAY be retrieved by Anybody.

Table 45 Locking SP - LockingInfo Table Preconfiguration

UID	Name	Version	EncryptSupport	MaxRanges	MaxReEncryptions	KeysAvailableCfg	AlignmentRequired	LogicalBlockSize	AlignmentGranularity	LowestAlignedLBA
00 00 08 01 00 00 00 01			Media Encryption	8[1]						

Note:

1. The MaxRanges column in Table 45 specifies the number of supported ranges and SHALL have a minimum of 8 ranges.

4.3.5.2 Locking (M)

Table 46 contains Optional rows designated with (O).
*LT1 means that the ActiveKey can be a K_AES_128 object reference (UID) or a K_AES_256 object reference (UID)
*LT2 means that only a limited set of LockOnReset values is required to be supported by Opal SSC SDs. Refer to section 4.3.5.2.2 for details.

Table 46 Locking SP - Locking Table Preconfiguration

UID	Name	CommonName	RangeStart	RangeLength	ReadLockEnabled	WriteLockEnabled	ReadLocked	WriteLocked	LockOnReset	ActiveKey	NextKey	ReEncryptState	ReEncryptRequest	AdvKeyMode	VerifyMode	ContOnReset	LastReEncryptLBA	LastReEncState	GeneralStatus
00 00 08 02 00 00 00 01	"Locking_GlobalRange"	""	0	0	F	F	F	F	Power Cycle *LT2	K_AES_128[256]_GlobalRange_Key *LT1									
00 00 08 02 00 03 00 01	"Locking_Range1"	""	0	0	F	F	F	F	Power Cycle *LT2	K_AES_128[256]_Range1_Key *LT1									
00 00 08 02 00 03 NN NN	"Locking_RangeNNNN"	""	0	0	F	F	F	F	Power Cycle *LT2	K_AES_128[256]_RangeNNNN_Key *LT1									

4.3.5.2.1 Geometry Reporting Feature Behavior

The following behaviors SHALL be implemented.

4.3.5.2.1.1 RangeStart Behavior

This column value defines the starting LBA value for this range. In non-Global Range rows, this column MAY be modifiable based on access control settings. Changes to this column are subject to the same constraints and checks defined for this column when rows of the `Locking` table are created (see [2]).

When processing a Set method or CreateRow method on the `Locking` table for a non-Global Range row, if:

a) the AlignmentRequired column in the `LockingInfo` table is TRUE;
b) RangeStart is non-zero; and
c) StartAlignment (see Figure 1) is non-zero,

then the method SHALL fail and return an error status code INVALID_PARAMETER.

Figure 1 – StartAlignment Calculation

StartAlignment = (RangeStart - LowestAlignedLBA) modulo AlignmentGranularity
where:
LowestAlignedLBA and AlignmentGranularity are columns in the `LockingInfo` table (see section 4.3.5.1)

4.3.5.2.1.2 RangeLength Behavior

This column value defines the quantity of contiguous LBAs for this LBA range (starting with the value defined in the RangeStart column). In non-Global Range rows, this column MAY be modifiable based on access control settings. Changes to this column are subject to the same constraints and checks defined for this column when rows of the `Locking` table are created (see [2]).

When processing a Set method or CreateRow method on the `Locking` table for a non-Global Range row, if:

a) the AlignmentRequired column in the `LockingInfo` table is TRUE;
b) RangeLength is non-zero; and
c) LengthAlignment (see Figure 2) is non-zero,

then the method SHALL fail and return an error status code INVALID_PARAMETER.

Figure 2 - LengthAlignment Calculation

If RangeStart is zero, then

LengthAlignment = (RangeLength - LowestAlignedLBA) modulo AlignmentGranularity

If RangeStart is non-zero, then

LengthAlignment = (RangeLength modulo AlignmentGranularity)

where:
LowestAlignedLBA and AlignmentGranularity are columns in the `LockingInfo` table (see section 4.3.5.1)

4.3.5.2.2 LockOnReset Restrictions

The TPer SHALL support the following LockOnReset column values:

a) { 0 } (i.e. Power Cycle); and
b) { 0, 3 } (i.e. Power Cycle and Programmatic).

Additionally, the TPer MAY support the following LockOnReset column values:

a) { 0, 1 } (i.e. Power Cycle and Hardware Reset); and

b) { 0,1, 3 } (i.e. Power Cycle, Hardware Reset and Programmatic).

4.3.5.3 MBRControl (M)

The MBRControl Table is defined in [2], and Table 47 defines the Preconfiguration Data for the MBRControl Table.

*MC1 means that only a limited set of DoneOnReset values is required to be supported by Opal SSC Storage Devices. Refer to section 4.3.5.3.1 for details.

Table 47 Locking SP - MBRControl Table Preconfiguration

UID	Enable	Done	DoneOnReset
00 00 08 03 00 00 00 01	False	**False**	**Power Cycle** *MC1

4.3.5.3.1 DoneOnReset Restrictions

The TPer SHALL support the following DoneOnReset column values:

a) { 0 } (i.e. Power Cycle); and
b) { 0, 3 } (i.e. Power Cycle and Programmatic).

Additionally, the TPer MAY support the following DoneOnReset column values:
a) { 0, 1 } (i.e. Power Cycle and Hardware Reset); and
b) { 0,1, 3 } (i.e. Power Cycle, Hardware Reset and Programmatic).

4.3.5.4 MBR (M)

The MBR minimum size SHALL be 128 MB (0x08000000).

The initial contents of the MBR table SHALL be vendor unique.

4.3.5.5 K_AES_128 or K_AES_256 (M)

At least one of the following tables Table 48 or Locking SP - K_AES_256 Table PreconfigurationTable 49 SHALL be supported.

Table 48 contains Optional rows designated with (O).
*K1 means that a field is indirectly writable using the GenKey Method.

Table 48 Locking SP - K_AES_128 Table Preconfiguration

UID	Name	CommonName	Key	Mode
00 00 08 05 00 00 00 01	"K_AES_128_GlobalRange_Key"		**VU** *K1	**VU**
00 00 08 05 00 03 00 01	"K_AES_128_Range1_Key"		**VU** *K1	**VU**
00 00 08 05 00 03 NN NN	"K_AES_128_RangeNNNN_Key"		**VU**	**VU**

UID	Name	CommonName	Key	Mode
(O)			*K1	

Table 49 Locking SP - K_AES_256 Table Preconfiguration

UID	Name	CommonName	Key	Mode
00 00 08 0600 00 00 01	"K_AES_256_GlobalRange_Key"		VU *K1	VU
00 00 08 06 00 03 00 01	"K_AES_256_Range1_Key"		VU *K1	VU
00 00 08 06 00 03 NN NN (O)	"K_AES_256_RangeNNNN_Key"		VU *K1	VU

4.3.6 Locking Template Methods

Refer to Section 4.3.1.5 for supported methods.

4.3.7 SD Read/Write Data Command Locking Behavior Interactions with Range Crossing

If an SD receives a read or write command that spans multiple Locking ranges and the Locking ranges are not locked, the SD SHALL either:

- Process the data transfer as defined in [2], if Range Crossing Behavior bit is set to zero (in Level 0 Discovery Opal SSC V2 Feature, see section 3.1.1.5)

  OR

- Terminate the command with “Other Invalid Command Parameter” as defined in [4],
  if Range Crossing Behavior bit is set to one (in Level 0 Discovery Opal SSC V2 Feature, see section 3.1.1.5).

4.3.8 Non Template Tables

4.3.8.1 DataStore (M)

The `DataStore` is a byte table. It can be used by the host for generic secure data storage. The `DataStore` table SHALL be at least 10MB in size (the `Table` table object that represents the `DataStore` table SHALL have a Rows column value of at least 0x00A00000). The access control for modification or retrieval of data in the table initially requires a member of the Admins class authority. These access control settings are personalizable. The Initial `DataStore` content value is VU.

5 Appendix – SSC Specific Features

5.1 Opal SSC-Specific Methods

5.1.1 Activate – Admin Template SP Object Method

`Activate` is an Opal SSC-specific method for managing the life cycle of SPs created in manufacturing (Manufactured SP), whose initial life cycle state is "Manufactured-Inactive". The following pseudo-code is the signature of the `Activate` Method (see [2] for more information).

```
SPObjectUID.Activate[ ]
=>
[ ]
```

`Activate` is an object method that operates on objects in the Admin SP's `SP` table. The TPer SHALL NOT permit `Activate` to be invoked on the SP objects of issued SPs.

Invocation of `Activate` on an SP object that is in the "Manufactured-Inactive" state causes the SP to transition to the "Manufactured" state. Invocation of `Activate` on an SP in any other life cycle state SHALL complete successfully provided access control is satisfied, and have no effect. The `Activate` method allows the TPer owner to "turn on" an SP that was created in manufacturing.

This method operates within a Read-Write session to the Admin SP. The SP SHALL be activated immediately after the method returns success if its invocation is not contained within a transaction.

In case of an "Activate Error" (see [4]) `Activate` SHALL fail with a status of FAIL.

The MethodID for `Activate` SHALL be 00 00 00 06 00 00 02 03.

5.1.1.1 Activate Support

Support for `Activate` within transactions is (N), and the behavior of `Activate` within transactions is out of the scope of this specification.

If the Locking SP was created in manufacturing, and its Original Factory State is Manufactured-Inactive (see section 5.2.2), support for `Activate` on the Locking SP's object in the `SP` table is Mandatory.

5.1.1.2 Side effects of Activate

Upon successful activation of an SP that was in the "Manufactured-Inactive" state, the following changes SHALL be made:

- The `LifeCycleState` column of SP's object in the Admin SP's `SP` table SHALL change to "Manufactured".
- The current SID PIN (C_PIN_SID) in the Admin SP is copied into the `PIN` column of Admin1's `C_PIN` credential (C_PIN_Admin1) in the activated SP. This allows for taking ownership of the SP with a known PIN credential.
- Any TPer functionality affected by the life cycle state of the SP based on the SP's templates is modified as defined in the appropriate Template reference section of [2], and as defined in the "State transitions for Manufactured SPs" section (see section 5.2.2.2) and "State behaviors for Manufactured SPs" section (see section 5.2.2.3) of this specification.

5.1.2 Revert – Admin Template SP Object Method

`Revert` is an Opal SSC-specific method for managing the life cycle of SPs created in manufacturing (Manufactured SP). The following pseudo-code is the signature of the `Revert` Method (see [2] for more information).

```
SPObjectUID.Revert[ ]
=>
[ ]
```

Revert is an object method that operates on objects in the Admin SP's SP table. The TPer SHALL NOT permit Revert to be invoked on the SP objects of issued SPs.

Invoking Revert on an SP object causes the SP to revert to its Original Factory State. This method allows the TPer owner (or TPer manufacturer, if access control permits and the Maker authorities are enabled) to remove the SP owner's ownership of the SP and revert the SP to its Original Factory State.

Invocation of Revert is permitted on Manufactured SPs that are in any life cycle state. Successful invocation of Revert on a Manufactured SP that is in the Manufactured-Inactive life cycle state SHALL have no effect on the SP.

This method operates within a Read-Write session to the Admin SP. The TPer SHALL revert the SP immediately after the method is successfully invoked outside of a transaction. If Revert is invoked on the Admin SP's object in the SP table, the TPer SHALL abort the session immediately after reporting status of the method invocation if invoked outside of a transaction. The TPer MAY prepare a CloseSession method for retrieval by the host to indicate that the session has been aborted.

The MethodID for Revert SHALL be 00 00 00 06 00 00 02 02.

5.1.2.1 Revert Support

Support for Revert within transactions is (N), and the behavior of Revert within transactions is out of the scope of this specification.

Support for Revert on the Admin SP's object in the SP table is Mandatory. (Note that the OFS of the Admin SP is Manufactured, see section 5.2.2).

If the Locking SP was created in manufacturing, support for Revert on the Locking SP's object in the SP table is Mandatory.

5.1.2.2 Effects of Revert

Upon successful invocation of the Revert method, the following changes SHALL be made:

- If the Locking SP is not in the "Manufactured-Inactive" life cycle state, then successful invocation of the Revert method on the Locking SP or Admin SP SHALL cause user data removal as defined by the ActiveDataRemovalMechanism (see Table 34) and cause the media encryption keys to be eradicated, which has the side effect of securely erasing all data in the User LBA portion of the SD.

- If the Locking SP is in the "Manufactured-Inactive" life cycle state, then successful invocation of the Revert method on the Locking SP SHALL NOT cause user data removal in the SD.

Interactions with interface commands during the processing of the Revert method are defined in [4].

If any TCG reset occurs prior to completing user data removal and the eradication of all media encryption keys in the SD, then the Revert operation SHALL be aborted and the Locking SP SHALL NOT revert to its Original Factory State.

Start of Informative Comment

If any TCG reset occurs during the processing of the Revert method, the result of user data removal is undefined and the TPer does not erase personalization of the Locking SP. For example, the PIN column value for each row in C_PIN table is unchanged.

End of Informative Comment

Upon completion of user data removal and the eradication of all media encryption keys in the SD, or if the Locking SP is in the "Manufactured-Inactive" life cycle state, the following changes SHALL be made:

- The row in the Admin SP's SP table that represents the invoked SP SHALL revert to its original factory values.
- The SP itself SHALL revert to its Original Factory State. While reverting to its Original Factory State, the TPer SHALL securely erase all personalization of the SP, and return personalized values to their Original Factory State values. The mechanism for erasure of personalization is implementation-specific.

- When Revert is successfully invoked on the SP object for the Admin SP (UID = 00 00 02 05 00 00 00 01), the entire TPer SHALL revert to its Original Factory State, including:
  - All Admin SP personalization with the exception of the PIN column value of the C_PIN_SID object. See section 5.1.2.2.1 for the effects of Revert upon the PIN column value of the C_PIN_SID object.
  - All issued SPs SHALL be deleted, and all Manufactured SPs SHALL revert to Original Factory State. Manufactured SPs in the “Manufactured-Inactive” life cycle state SHALL NOT be affected.
- Any TPer functionality affected by the life cycle state of the SP based on the templates incorporated into it is modified as defined in the appropriate Template reference section of [2], and as defined in the “State transitions for Manufactured SPs” section (see section 5.2.2.2) and “State behaviors for Manufactured SPs” section (see section 5.2.2.3) of this specification.

Start of Informative Comment

Unless already in the Manufactured-Inactive life cycle state, reverting the Locking SP will cause the media encryption keys to be eradicated, which has the side effect of securely erasing all data in the User LBA portion of the Storage Device.

End of Informative Comment

5.1.2.2.1 Effects of Revert on the PIN Column Value of C_PIN_SID

When Revert is successfully invoked on the SP object for the Admin SP (UID = 00 00 02 05 00 00 00 01), the PIN column value of the C_PIN_SID object SHALL be affected as follows:

1. If the SID authority has never been successfully authenticated, then the C_PIN_SID PIN column SHALL remain at its current value.
2. If the SID authority has previously been successfully authenticated, then:

   a) If the value of the “Behavior of C_PIN_SID PIN upon TPer Revert” field in the Opal SSC V2 Feature Descriptor is 0x00, then the C_PIN_SID PIN column SHALL be set to the PIN column value of the C_PIN_MSID object. Additionally, the “Initial C_PIN_SID PIN Indicator” field SHALL be set to 0x00 upon completion of the Revert.
   b) If the value of the “Behavior of C_PIN_SID PIN upon TPer Revert” field in the Opal SSC V2 Feature Descriptor is not 0x00, then the C_PIN_SID PIN column SHALL be set to a vendor unique (VU) value.

Start of Informative Comment

In the case where the “Initial C_PIN_SID PIN Indicator” and “Behavior of C_PIN_SID PIN upon TPer Revert” fields are both 0x00, the above rules for Revert are backward compatible with Opal v1.00.

End of Informative Comment

5.1.2.3 Interrupted Revert

The Revert method and complete implementation of necessary background operations MAY be aborted due to any reset condition, including power loss.

When interrupted, the Data Removal Operation Interrupted bit SHALL be set to one in the Level 0 Discovery – Supported Data Removal Mechanism feature descriptor appropriately as defined in section 3.1.1.6.2.

Further, the return status value of the Revert method does not mean that all necessary operations, such as the background deallocate, or trim, or un-map are complete.

5.1.3 RevertSP – Base Template SP Method

RevertSP is an Opal SSC-specific method for managing the life cycle of an SP, if it was created in manufacturing (Manufactured SP). The following pseudo-code is the signature of the RevertSP Method (see [2] for more information).

```
ThisSP.RevertSP[ KeepGlobalRangeKey = boolean ]
=>
```

```
[ ]
```

`RevertSP` is an SP method in the Base Template.

Invoking `RevertSP` on an SP SHALL cause it to revert to its Original Factory State. This method allows the SP owner to relinquish control of the SP and revert the SP to its Original Factory State.

This method operates within a Read-Write session to an SP. The TPer SHALL revert the SP immediately after the method is successfully invoked outside of a transaction. Upon completion of reverting the SP, the TPer SHALL report status of the method invocation if invoked outside of a transaction, and then immediately abort the session. The TPer MAY prepare a `CloseSession` method for retrieval by the host to indicate that the session has been aborted.

The MethodID for `RevertSP` SHALL be 00 00 00 06 00 00 00 11.

5.1.3.1 RevertSP Support

Support for `RevertSP` within transactions is (N), and the behavior is out of the scope of this document.

If the Locking SP was created in manufacturing, support for `RevertSP` on the Locking SP is Mandatory.

5.1.3.2 KeepGlobalRangeKey parameter (Locking Template-specific)

The Optional **KeepGlobalRangeKey** parameter is a Locking Template-specific parameter. This parameter provides a mechanism for the Locking SP to be “turned off” without eradicating the media encryption key for the Global Locking Range. This allows the Locking SP to be disabled without causing removal of the user data associated with the Global Locking Range.

When this parameter is present and set to True, the TPer SHALL NOT erase data associated with the Global Locking Range after the Locking SP transitions to the “Manufactured-Inactive” state even if the valid value is set to the ActiveDataRemovalMechanism parameter in DataRemovalMechanism table.

If the Global Range is either Read Unlocked or Write Unlocked at the time of invocation of `RevertSP`, then the TPer SHALL comply with the request to keep the user data associated with the Global locking range and the Global Range’s-media encryption key.

If the Global Range is Read Locked and Write Locked then invocation of the `RevertSP` method with the **KeepGlobalRangeKey** parameter set to True SHALL fail with status FAIL, and the SP SHALL NOT change life cycle states.

If the Locking SP was created in manufacturing, support for the **KeepGlobalRangeKey** parameter is Mandatory for the Locking SP.

The parameter number for **KeepGlobalRangeKey** SHALL be 0x060000.

5.1.3.3 Effects of RevertSP

Upon successful invocation of the `RevertSP` method, the following changes SHALL be made:

- If the **KeepGlobalRangeKey** parameter is not present or set to False, then successful invocation of the `RevertSP` method on the Locking SP or Admin SP SHALL cause user data removal as defined by the ActiveDataRemovalMechanism (see Table 34) and cause the media encryption keys to be eradicated, which has the side effect of securely erasing all data in the User LBA portion of the SD.
- If the **KeepGlobalRangeKey** parameter is set to True, then successful invocation of the `RevertSP` method on the Locking SP SHALL cause user data removal in the SD all media encryption keys to be eradicated except for the Global Range’s media encryption key (K_AES_{128,256}_GlobalRange_Key).

Interactions with interface commands during the processing of the `RevertSP` method are defined in [4].

If any TCG reset occurs prior to completing user data removal and the eradication of media encryption keys in the SD, then the operation SHALL be aborted and the Locking SP SHALL NOT revert to its Original Factory State.

Start of Informative Comment

If any TCG reset occurs during the processing of the `RevertSP` method, the result of user data removal is undefined.

End of Informative Comment

Upon completion of user data removal and the eradication of media encryption keys in the SD, the following changes SHALL be made:

- The row in the Admin SP's `SP` table that represents the Locking SP SHALL revert to its original factory value.
- The Locking SP itself SHALL revert to its Original Factory State. While reverting to its Original Factory State, the TPer SHALL erase all personalization of the SP, and return the personalized values to their Original Factory State values. The mechanism for erasure of personalization implementation-specific.
- Any TPer functionality affected by the life cycle state of the SP based on the templates incorporated into it is modified as defined in the appropriate Template reference section of [2], and as defined in the "State transitions for Manufactured SPs" section (see section 5.2.2.2) and "State behaviors for Manufactured SPs" section (see section 5.2.2.3) of this specification.

Start of Informative Comment

Reverting the Locking SP will cause the media encryption keys to be eradicated (except for the GlobalRange key if the **KeepGlobalRangeKey** parameter is present and set to True), which has the side effect of securely erasing all data in the User LBA portion of the Storage Device.

End of Informative Comment

5.1.3.4 Interrupted RevertSP

The `RevertSP` method and complete implementation of the necessary background operations MAY be aborted due to any reset condition, including power loss.

When interrupted, the Data Removal Operation Interrupted bit SHALL be set to one in the Level 0 Discovery – Supported Data Removal Mechanism feature descriptor appropriately as defined in section 3.1.1.6.2.

Further, the return status value of the `RevertSP` method does not mean that all necessary operations such as the data removal operation are complete.

5.2 Life Cycle

5.2.1 Issued vs. Manufactured SPs

5.2.1.1 Issued SPs

For Opal SSC-compliant TPers that support issuance, refer to [2] for the life cycle states and life cycle management.

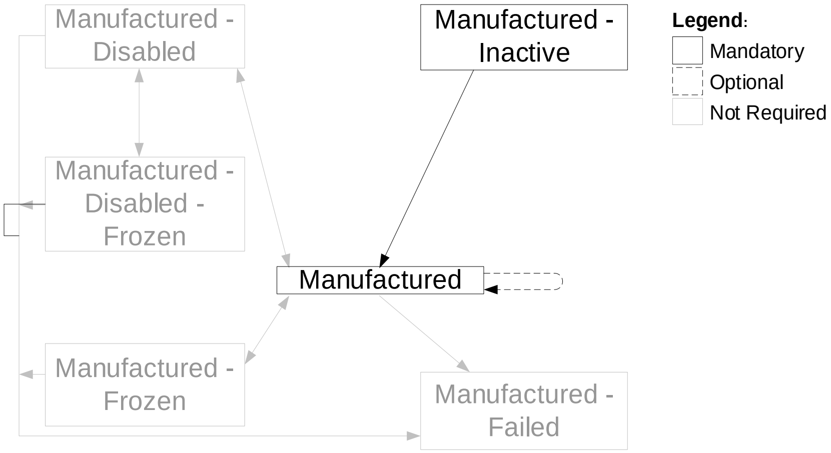
5.2.1.2 Manufactured SPs

Opal SSC-compliant SPs that are created in manufacturing (Manufactured SPs) SHALL NOT have an implementation-specific life cycle, and SHALL conform to the life cycle defined in section 5.2.2.

5.2.2 Manufactured SP Life Cycle States

The state diagram for Manufactured SPs is shown in Figure 3.

Figure 3 - Life Cycle State Diagram for Manufactured SPs



Additional state transitions may exist depending on the states supported by the SD and the SP's Original Factory State. Invoking `Revert` or `RevertSP` (see sections 5.1.2 and 5.1.2.3) on the SP will cause the SP to transition back to its Original Factory State.

The Original Factory State of the Admin SP SHALL be Manufactured. The only state that is Mandatory for the Admin SP is Manufactured.

If the Locking SP is a Manufactured SP, its Original Factory State SHALL be Manufactured-Inactive.

Support for Locking SP states of Manufactured and Manufactured-Inactive are mandatory.

The other states in the state diagram are beyond the scope of this document.

5.2.2.1 State definitions for Manufactured SPs

1. **Manufactured-Inactive**: This is the Original Factory State for SPs that are created in manufacturing, where it is not desired for the functionality of that SP to be active when the TPer is shipped. All templates that exist in an SP that is in the Manufactured-Inactive state SHALL be counted in the `Instances` column of the appropriate objects in the Admin SP's `Template` table. Sessions cannot be opened to SPs in the Manufactured-Inactive state. Only SPs whose Original Factory State was Manufactured-Inactive can return to the Manufactured-Inactive state.

   If the Locking SP is a Manufactured SP, support for the Manufactured-Inactive state is Optional for the Locking SP.

2. **Manufactured**: This is the standard operational state of a Manufactured SP, and defines the initial required access control settings of an SP based on the Templates incorporated into the SP, prior to personalization.

   The Manufactured state is Mandatory for the Admin SP.

   If the Locking SP is a Manufactured SP, support for the Manufactured state is Mandatory for the Locking SP.

5.2.2.2 State transitions for Manufactured SPs

The following sections describe the Mandatory and Optional state transitions for Opal SSC-compliant Manufactured SPs.

For the Admin SP, the only transition for which support is mandatory is "ANY STATE to ORIGINAL FACTORY STATE" (see section 5.2.2.2.2). As the only mandatory state for the Admin SP is Manufactured, the only mandatory transition is from Manufactured to Manufactured with the side effect of reverting the entire TPer to its Original Factory State. See section 5.1.2 for details.

If the Locking SP is a Manufactured SP, support for the "ANY STATE to ORIGINAL FACTORY STATE" transition (see section 5.2.2.2.2) is Mandatory. Specifically, support for the transition from Manufactured to either Manufactured-

Inactive or Manufactured is Mandatory, depending on the Locking SP's Original Factory State. This transition is accomplished via the Revert or RevertSP method (see sections 5.1.2 and 5.1.2.3).

If the Locking SP's Original Factory State is Manufactured-Inactive, then support for the "Manufactured-Inactive to Manufactured" transition (see section 5.2.2.2.1) is Mandatory. This transition is accomplished via the Activate method (see section 5.1.1).

5.2.2.2.1 Manufactured-Inactive to Manufactured

Triggers:

- The Activate method (see section 5.1.1) is successfully invoked on the SP's object in the Admin SP's SP table.

Side effects:

- The value in the LifeCycleState column of the SP's object in the Admin SP's SP table changes to Manufactured.
- The current SID PIN (C_PIN_SID) in the Admin SP is copied into the PIN column of Admin1's C_PIN credential (C_PIN_Admin1) in the activated SP. This allows taking ownership of the SP with a known PIN credential.
- Any functionality enabled by the templates incorporated into the SP becomes active.

When the Locking SP transitions from the Manufactured-Inactive state to the Manufactured state (via invocation of the Activate method), the SD SHALL NOT destroy any user data.

5.2.2.2.2 ANY STATE to ORIGINAL FACTORY STATE

Triggers:

- Revert or RevertSP is successfully invoked on the SP.

Side effects:

- The value in the LifeCycleState column of the SP's object in the Admin SP's SP table changes to the value of the SP's Original Factory State.
- The SP itself reverts to its Original Factory State, as described in sections 5.1.2 and 5.1.3.
- If the SP's Original Factory State was Manufactured-Inactive, any functionality enabled by the templates incorporated into the SP becomes inactive.

5.2.2.3 State behaviors for Manufactured SPs

5.2.2.3.1 Manufactured-Inactive

Any functionality enabled by the templates incorporated into the SP is inactive in this state. Sessions cannot be opened to SPs in this state.

When the Locking SP is in the Manufactured-Inactive state, the Locking SP's management of the SD's locking and media encryption features SHALL be disabled.

5.2.2.3.2 Manufactured

Behavior of an SP in the Manufactured state is identical to the behavior of an SP in the Issued state, as described in [2].

When the Locking SP is in the Manufactured state, the Locking SP's management of the SD's locking and media encryption features SHALL be enabled.

5.2.3 Type Table Modification

In order to accommodate the additional life cycle states defined in this specification, the definition of the life_cycle_state type is changed from [2] to that described in Table 50:

Table 50 LifeCycle Type Table Modification

UID	Name	Format	Size	Description
00 00 00 05 00 00 04 05	life_cycle_state	Enumeration_Type, 0, 15		Used to represent the current life cycle state. The valid values are: 0 = issued, 1 = issued-disabled, 2 = issued-frozen, 3 = issued-disabled-frozen, 4 = issued-failed, 5-7 = reserved, 8 = manufactured-inactive, 9 = manufactured, 10 = manufactured-disabled, 11 = manufactured-frozen, 12 = manufactured-disabled-frozen, 13 = manufactured-failed, 14-15 = reserved

5.3 Byte Table Access Granularity

Start of Informative Comment

While the general architecture defined in [2] allows data to be written into byte tables starting at any arbitrary byte boundary and with any arbitrary byte length, certain types of storage SDs work more efficiently when data is written aligned to a larger block boundary. This section defines extensions to [2] that allow a SD to report the restrictions that it enforces when the host invokes the `Set` method on byte tables.

End of Informative Comment

5.3.1 Table Table Modification

In order to allow a SD to report its mandatory and recommended data alignment restrictions when accessing byte tables, the `Table` table SHALL contain the additional columns shown in Table 51.

The mandatory and recommended data alignment restrictions do not apply to Object Tables.

Table 51 Table Table Additional Columns

Column Number	Column Name	IsUnique	Column Type
0x0D	MandatoryWriteGranularity		uinteger_4
0x0E	RecommendedAccessGranularity		uinteger_4

5.3.1.1 MandatoryWriteGranularity

This column is used to report the granularity that the SD enforces when the host invokes the `Set` method on byte tables.

This column SHALL NOT be modifiable by the host.

5.3.1.1.1 Object Tables

For rows in the `Table` table that pertain to object tables, the value of the MandatoryWriteGranularity column SHALL be zero.

5.3.1.1.2 Byte Tables

For rows in the `Table` table that pertain to byte tables, the MandatoryWriteGranularity column indicates the mandatory access granularity (in bytes) for the `Set` method for the table described in these rows of the `Table` table. The

MandatoryWriteGranularity column indicates the alignment requirement for both the access start offset (the Where parameter) and length (number of bytes in the Values parameter).

The value of the MandatoryWriteGranularity column SHALL be less than or equal to the value in the RecommendedAccessGranularity column in the same row of the `Table` table.

The value of MandatoryWriteGranularity SHALL be less than or equal to 8192.

When the host invokes the `Set` method on a byte table, if ValidMandatoryGranularity (see Figure 4) is False, then the method SHALL fail with status INVALID_PARAMETER.

If the TPer does not have a requirement on mandatory alignment for the byte table described in a row of the `Table` table, then its MandatoryWriteGranularity column SHALL be set to one.

Figure 4 - ValidMandatoryGranularity definition

```
For the Set method:
  ValidMandatoryGranularity is True if
          a) (x modulo MandatoryWriteGranularity) = 0

             and

          b) (y modulo MandatoryWriteGranularity) = 0

where:
  x =  the start offset of the Set method
         (i.e., the value of the Where parameter)
  y = the number of data bytes being set
         (i.e., the length of the Values parameter)
```

5.3.1.2 RecommendedAccessGranularity

This column is used to report the granularity that the SD recommends when the host invokes the `Set` or `Get` method on byte tables.

This column SHALL NOT be modifiable by the host.

5.3.1.2.1 Object Tables

For rows in the `Table` table that pertain to object tables, the value of the RecommendedAccessGranularity column SHALL be zero.

5.3.1.2.2 Byte Tables

For rows in the `Table` table that pertain to byte tables, the RecommendedAccessGranularity column indicates the recommended access granularity (in bytes) for the `Set` and `Get` method for the table described in these rows of the `Table` table. The RecommendedAccessGranularity column indicates the alignment of data for the `Set` and `Get` method that allows for optimal `Set`/`Get` performance.

If the TPer does not have a recommended alignment for the byte table described in a row of the `Table` table, then its RecommendedAccessGranularity column SHALL be set to one.

When the host invokes the `Set` method on a byte table, if ValidRecommendedGranularity (see Figure 5) is False, then the performance of the TPer MAY be reduced when processing the method.

Figure 5 - ValidRecommendedGranularity definition for Set

For the `Set` method:
ValidRecommendedGranularity is True if
a) (x modulo RecommendedAccessGranularity) = 0

and

b) (y modulo RecommendedAccessGranularity) = 0

where:
x = the start offset of the `Set` method
(i.e., the value of the Where parameter)
y = the number of data bytes being set
(i.e., the length of the Values parameter)

When the host invokes the `Get` method on a byte table, if ValidRecommendedGranularity (see Figure 6) is False, then the performance of the TPer MAY be reduced when processing the method.

Figure 6 - ValidRecommendedGranularity definition for Get

For the `Get` method:
ValidRecommendedGranularity is True if
a) (x modulo RecommendedAccessGranularity) = 0

and

b) (y modulo RecommendedAccessGranularity) = 0

where:
x = the start offset of the `Get` method
(i.e., the value of the startRow component of the Cellblock parameter)
y = the number of data bytes being retrieved
(i.e., the difference of the endRow and startRow components of the Cellblock parameter, plus one)

5.4 Examples of Alignment Geometry Reporting

Figure 7 illustrates reporting for a typical legacy SD where there is one logical block per physical block on the media.

Figure 7 - Example: AlignmentGranularity=1, Lowest Aligned LBA=0

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Alignment Granularity																			

Figure 8 illustrates geometry for a SD where there are 8 logical blocks per physical block (e.g., a 4K physical block) and the first logical block is aligned at the beginning of the first physical block.

Figure 8 - Example: AlignmentGranularity=8, Lowest Aligned LBA=0

<table>
<tr><td><b>0</b></td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td></tr>
<tr><td colspan="8">AlignmentGranularity</td><td colspan="8">AlignmentGranularity</td><td colspan="4">. . .</td></tr>
</table>

Figure 9 illustrates geometry for a SD where there are 8 logical blocks per physical block (e.g., a 4K physical block) and LBA=1 is the first logical block that is aligned at the beginning of a physical block

Figure 9 - Example: AlignmentGranularity=8, Lowest Aligned LBA=1

<table>
<tr><td colspan="7"></td><td>0</td><td><b>1</b></td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td></tr>
<tr><td colspan="8">AlignmentGranularity</td><td colspan="8">AlignmentGranularity</td><td colspan="4">. . .</td></tr>
</table>

Figure 10 illustrates geometry for a SD where there are 2000 logical blocks per physical block and LBA=1234 is the first logical block that is aligned at the beginning of a physical block.

Figure 10 - Example: AlignmentGranularity=2000, Lowest Aligned LBA=1234

<table>
<tr><td></td><td>0</td><td>. . .</td><td>1230</td><td>1231</td><td>1232</td><td>1233</td><td><b>1234</b></td><td></td><td>. . .</td><td></td><td>3233</td><td>3234</td><td>. . .</td></tr>
<tr><td colspan="7">AlignmentGranularity</td><td colspan="5">AlignmentGranularity</td><td colspan="2">. . .</td></tr>
</table>