

TCG Storage Opal SSC: Test Cases Specification

Specification Version 2.00

Revision 1.00

March 16, 2016

Final

Contact: admin@trustedcomputinggroup.org

TCG

TCG PUBLISHED

Copyright © TCG 2016

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.

This document is copyrighted by Trusted Computing Group (TCG), and no license, express or implied, is granted herein other than as follows: You may not copy or reproduce the document or distribute it to others without written permission from TCG, except that you may freely do so for the purposes of (a) examining or implementing TCG specifications or (b) developing, testing, or promoting information technology standards and best practices, so long as you distribute the document with these disclaimers, notices, and license terms.

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.

Table of Contents

1. Introduction	1
1.1 Document Purpose and Scope	1
1.2 Intended Audience	1
1.3 Document References	1
1.3.1 Key Words and Symbols	1
1.3.2 Terminology	2
2. Opal SSC Test Cases Outline	4
2.1 Overview	4
2.2 Test Case Description	4
2.2.1 Notes	4
2.2.2 Prerequisites	4
2.2.3 Test Sequence	4
2.2.4 Expected Response	4
3. Common Baseline Conditions and Test Criteria	5
3.1 Minimum Test Requirements	5
3.2 Opal SSC 2.00 and 2.01 Specific Requirements	6
4. Use Case Test Cases	7
4.1 Introduction	7
4.2 Common Prerequisites	7
4.3 Level 0 Discovery	7
4.4 Properties	8
4.5 Taking Ownership of an SD	9
4.6 Activate Locking SP when in Manufactured-Inactive State	10
4.7 Configuring Authorities	11
4.8 Configuring Locking Objects (Locking Ranges)	13
4.9 Unlocking Ranges	14
4.10 Erasing Ranges	15
4.11 Using the DataStore Table	16
4.12 Enable MBR Shadowing	17
4.13 MBR Done	18
4.14 Revert the Locking SP using SID, with Locking SP in Manufactured state	18
4.15 Revert the Admin SP using SID, with Locking SP in Manufactured-Inactive state	19
4.16 Revert the Admin SP using SID, with Locking SP in Manufactured state	20
4.17 Revert Admin SP using Admin1, with Locking SP in Manufactured state – Opal SSC 2.00 and 2.01	21
4.18 Revert Admin SP using PSID, with Locking SP in Manufactured state	22
5. Specific Functionality	23
5.1 Common Prerequisites	23

5.2	Transaction	23
5.3	IF-RECV Behavior Tests	25
5.3.1	TPer Response to IF-RECV when in Awaiting IF-SEND	25
5.3.2	TPer Response to IF-RECV with Insufficient Transfer Length	25
5.4	TryLimit	26
5.5	Tries Reset	27
5.6	Tries Reset on Power Cycle	28
5.7	Next	30
5.8	Host Session Number (HSN)	31
5.9	RevertSP on Locking SP KeepGlobalRangeKey Parameter Effects	31
5.10	Range Alignment Verification – Opal SSC 2.00 and 2.01	33
5.11	Byte Table Access Granularity – Opal SSC 2.00 and 2.01	34
5.12	Stack Reset	34
5.13	TPer Reset – Opal SSC 2.00 and 2.01	35
5.14	Authenticate – Opal SSC 2.0 and 2.01	36
5.15	Session Abort	36
5.16	Random	37
5.17	CommonName – Opal SSC 2.00 and 2.01	37
5.18	DataStore Table – Opal SSC 2.00 and 2.01	38
5.19	Range Crossing Behavior	39
6.	<i>Error Test Cases</i>	41
6.1	Common Prerequisites	41
6.2	Native Protocol Read/Write Locked Error Responses	41
6.3	General – IF-SEND/IF-RECV Synchronous Protocol	42
6.4	Invalid ComPacket Header Length Field	42
6.5	Invalid SessionID - Regular Session	43
6.6	Unexpected Token Outside of Method – Regular Session	43
6.7	Unexpected Token in Method Header – Regular Session	44
6.8	Unexpected Token Outside of Method – Control Session	45
6.9	Unexpected Token in the Method Parameter List – Control Session	45
6.10	Exceeding Transaction Limit	46
6.11	Invalid Invoking ID - Get	46
6.12	Invalid Invoking ID – Non-Get	47
6.13	Authorization	48
6.14	Exceed TPer Properties – Regular Session	48
6.15	Exceed TPer Properties – Regular Session	49
6.16	Exceed TPer Properties – Control Session	50
6.17	Overlapping Locking Ranges	50
6.18	Invalid Type	51
6.19	RevertSP – GlobalRange Locked	51

6.20	Activate / ATA Security Interaction _____	52
6.21	StartSession on Inactive Locking SP _____	53
6.22	StartSession with Incorrect HostChallenge _____	53
6.23	Multiple Sessions _____	54

Tables

Table 1 Key Words.....	1
Table 2 Symbols	2
Table 3 Terminology	2
Table 4 IF-SEND/RECV Security Protocol=1 Command Field.....	5
Table 5 IF-SEND Security Protocol=1 Command Payload.....	6

1. Introduction

1.1 Document Purpose and Scope

This document defines test cases specific to the Opal SSC 1.00, 2.00 and 2.01 specifications. They are intended to provide guidance when testing the functionality of an SD. The test cases are based upon the requirements described in [1] [2] [3] [4] [5].

Not every feature or capability within those specifications is included in this document for testing. The test cases are driven by baseline capabilities of the Opal SSC, and by SD responses that can be verified by functional testing and are representative of expected use cases.

The test cases do not include any compatibility testing between Opal SSC versions.

1.2 Intended Audience

The intended audience for this document is SD manufacturers and software developers that may wish to tie SDs into trusted platforms, as well as manufacturers and developers of other components that intend to bind to trusted SDs. This document is also intended as a reference for test suite vendors.

This document assumes familiarity and working knowledge of [1] [2] [3] [4] [5].

1.3 Document References

- [1] TCG Storage Architecture Core Specification, Version 2.00
- [2] TCG Storage Interface Interactions Specification, Version 1.04
- [3] TCG Storage Security Subsystem Class: Opal, Version 1.00
- [4] TCG Storage Security Subsystem Class: Opal, Version 2.00
- [5] TCG Storage Security Subsystem Class: Opal, Version 2.01
- [6] Internet Engineering Task Force (IETF), "Key words for use in RFCs to Indicate Requirement Levels" (RFC 2119)
- [7] [INCITS T13/2015-D], "Information technology - ATA/ATAPI Command Set – 2 (ACS-2)"

1.3.1 Key Words and Symbols

Key words are used to signify the requirements in the specification. The key words "SHALL," "SHALL NOT," "MAY," and "OPTIONAL" are used in this document in the same manner as [1]. These key words are to be interpreted as described in [6].

Table 1 Key Words

Term	Definition
<*_PASSWORD>	32 byte hex value used as a PIN column value for the C_PIN object associated with the noted authority.
ARBITRARILY_VARYING	Refers to a value that varies between executions in an arbitrary way determined by the Test Suite Vendor.
ACTIVATE_THE_LOCKING_SP	Change the life cycle state of the Locking SP in a TPer from Manufactured-Inactive to Manufactured. See [2]
CLOSE_SESSION	The host transmits an End of Session token.

Term	Definition
ENABLE <AuthorityName>	Invoke the Set method to set Enabled column value to TRUE for the noted authority object.
EXPECTED_RESPONSE	See 2.2.4
FAIL FAILS	Expected failure of one or more test sequence steps.
MAGIC_PATTERN	A data sequence used in some of the test cases. It has a value of 0x306F0A4A571DC563, and is always aligned with the first byte of each logical block. This value was arbitrarily selected to be distinguishable as data that had been intentionally written by the host application.
SET_PASSWORD_FOR <C_PIN object name>	Invoke Set method on the PIN column of the noted C_PIN object to the value provided in the test sequence step.
SUCCEED SUCCEEDS	Test Sequence step(s) result in the appropriate response(s) as described in [1] [2] [3] [4] [5].
USER_DATA	Data that may be transferred between the host and the TPer using READ commands and WRITE commands.

Table 2 Symbols

Symbol	Definition
=	Equals/Equivalence
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to
<>	Is not equal to
-	Minus
+	Plus
%	Modulo

1.3.2 Terminology

Table 3 Terminology

Term	Definition
Host (or host)	An entity that initiates IF-SEND or IF-RECV to a TPer
IF-RECV	An interface command used to transmit data from the host to the TPer. See [1]

Term	Definition
IF-SEND	An interface command used by the host to retrieve data from TPer. See [1]
Original Factory State (OFS)	The original state of an SP in a TPer when shipped from the manufacturer's factory. See [2] [3] [4]
SWG	Storage Work Group. A Work Group of the Trusted Computing Group
SD	The Storage Device
target device	A TPer that is tested by a test suite in the test cases in this document.
Test Suite	Software that performs the indicated test sequences of each test.
TPer	A Trusted Peripheral. An entity that implements TCG SWG SP(s) and responds to an IF-SEND or an IF-RECV initiated by a Host. See [1]
Transfer Length	The Transfer Field of IF-SEND or IF-RECV (see [1]) or Transfer Length field value
user data	Data that may be transferred between the host and the TPer using READ commands and WRITE commands.

2. Opal SSC Test Cases Outline

2.1 Overview

Each test case description contains four components: Notes, Prerequisites, Test Sequence and Expected Response. Expected Response describes the expected behavior(s) of the target device in each test. Prerequisites define the initial conditions that have to be met prior to performing the test. Notes provide informative text relating to the test for context. Details of these four components are described in 2.2.

The majority of tests are contained in two areas: Section 4: Use Case Test Cases and Section 5: Specific Functionality. Additionally, Section 3 outlines data handling requirements for Test Suite vendors and Section 6 details required test cases for error conditions.

Test cases in Section 4 are required to be performed in sequential order.

2.2 Test Case Description

2.2.1 Notes

The Notes section is informative text. It contains any information pertinent to the test being performed. This component may not be populated for every test case.

2.2.2 Prerequisites

Sections 4, 5, and 6 include a set of common prerequisites for each section that SHALL be met prior to performing any test in that section. Additionally, each test case within a section may have prerequisites specific to that test that SHALL be met prior to performing the specific test. If there are no prerequisites required for a specific test case, this area states 'None' and the test begins with the Test Sequence criteria.

2.2.3 Test Sequence

The Test Sequence includes the required steps, in sequential order, that SHALL be performed to obtain the Expected Response for a given test. Test Sequences may include different steps for [3] [4] [5].

2.2.4 Expected Response

Expected Response describes the expected behavior(s) of the target device under the Prerequisites and Test Sequence condition(s). All the expected responses are defined in [1] [2] [3] [4] [5] [7].

The descriptions in Expected Response may contain different responses for [3] [4] [5].

3. Common Baseline Conditions and Test Criteria

3.1 Minimum Test Requirements

The Test Suite SHALL:

- a) utilize Synchronous Interface Communications capability (See [1]) for host to TPer communications
- b) comply with IF-SEND(s) and IF-RECV(s) command field values described in Table 4
- c) comply with IF-SEND payload field values described in Table 5
- d) contain a payload that SHALL NOT cause errors or state changes within the TPer (e.g. invocation of the Properties method) for tests that require examining the Interface Command Parameters or ComPacket/Packet/Subpacket headers with values other than described above
- e) utilize Read-Write sessions for Regular sessions
- f) adhere to the TPer communications capabilities as reported in the Properties method response unless specifically required to do otherwise for a specific test
- g) use the Extended ComID value provided under level 0 Discovery
- h) use the Host Session Number (HSN) 0x00000001, except in the specific Host Session Number (HSN) test defined in section 5.8
- i) have a Packet.SeqNumber of 0s for communications sent to the TPer

The Test Suite SHALL NOT:

- a) send empty atoms unless specifically required to do so for a test
- b) utilize Buffer Management capability (See [1])
- c) utilize ACK/NAK capability (See [1])

For invocations of IF-RECV tests, the TPer is in the Awaiting IF-RECV state for a ComID:

- a) when the ComPacket header 'OutstandingData' field = 1 the Test Suite SHALL re-issue an IF-RECV until the TPer returns a ComPacket header that does not satisfy the condition, or
- b) when the ComPacket 'OutstandingData' field = <total data available>; and the 'MinTransfer' field = <minimum request length required to transfer a packet>, the Test Suite SHALL issue another IF-RECV with greater value of Transfer Length than the previous until the TPer returns a response that does not satisfy the conditions
- c) when the TPer response contains a Subpacket and the ComPacket 'OutstandingData' field = <additional bytes available, not including the data transferred in the current ComPacket>; and the 'MinTransfer' field = <minimum request required to transfer the next packet>, the Test Suite SHALL issue and IF-RECV until the TPer returns a response that does not satisfy the above conditions

Table 4 IF-SEND/RECV Security Protocol=1 Command Field

Field	Value
Security Protocol	1
Security Protocol Specific	any static ComID the TPer supports and as reported by the Opal SSC Feature Descriptor
Transfer Length	the minimum value necessary to transfer a ComPacket

Table 5 IF-SEND Security Protocol=1 Command Payload

Field	Value
ComPacket Header	
Reserved	all-0s
ComID	the same value as the Security Protocol Specific field in the IF-SEND
ComID Extension	all-0s
OutstandingData	all-0s
MinTransfer	all-0s
Length	a value which satisfies the following conditions: a) multiple-of-4; b) does not exceed (the TPer's MaxComPacketSize – 20); and c) indicates its payload contains exactly one Packet
Packet Header	
Session	a) all-0s for Control session; or b) the session number of the session that was successfully started by a StartSession() and a SyncSession() for Regular session
SeqNumber	all-0s
Reserved	all-0s
AckType	all-0s
Acknowledgement	all-0s
Length	a value which satisfies the following conditions: a) multiple-of-4; b) does not exceed (the TPer's MaxPacketSize – 24); and c) indicates its payload contains exactly one Subpacket and one Pad field, if necessary
Subpacket Header	
Reserved	all-0s
Kind	all-0s
Length	such value that is exactly the length of token stream the host is sending to the TPer
Pad	
	all-0s (and its length is 0 to 3)

3.2 Opal SSC 2.00 and 2.01 Specific Requirements

If SID is not MSID, the Test Suite vendor SHALL obtain the value of SID from SD vendor prior to conducting tests.

4. Use Case Test Cases

4.1 Introduction

Test cases in this section relate to use case scenarios that apply to general SD functionality. Tests in this section SHALL be performed in sequential order. Unless otherwise specified within a test case, the expected result of each step is that the step SHALL SUCCEED.

4.2 Common Prerequisites

Unless otherwise noted, the following set of prerequisites apply for each test in this section:

1. Synchronous Protocol state machine for all ComIDs is in “Awaiting IF-SEND” state
2. The Locking SP is in Manufactured state
3. The values of any credentials used are known
4. All StartSession method HostChallenge parameters use the current C_PIN object’s PIN column value for the Authority used in the HostSigningAuthority parameter
5. All sessions are Read-Write sessions
6. No open sessions exist at the start of the Test Sequence

4.3 Level 0 Discovery

Notes

Begin Informative Content

This test includes the sequence of operations required to determine if an SD supports Opal SSC 1.00, 2.00, or 2.01.

After completing this test, record the COM ID value for use in later tests.

End Informative Content

Prerequisites

None

Test Sequence

- 1) Issue an IF-RECV Level 0 Discovery with the following conditions:
 - a. Security Protocol = 1
 - b. Security Protocol Specific = 0x0001
 - c. Transfer Length is a value large enough to retrieve the entire response data of Level 0 Discovery

Expected Response

- 1) Step #1 SUCCEEDS

- 2) The SD returns the following values for Level 0 Discovery:
 - a. TPer Feature
 - i. Feature Code = 0x0001
 - ii. Streaming Supported = 1
 - iii. Sync Supported = 1
 - b. Locking Feature
 - i. Feature Code = 0x0002
 - ii. Media Encryption = 1
 - iii. Locking Supported = 1
- 3) The SD returns the following values for Opal SSC 1.00:
 - a. Opal SSC 1.00 Feature
 - i. Feature Code = 0x0200
 - ii. Number of COM IDs \geq 1
- 4) The SD returns the following values for Opal SSC 2.00 or 2.01:
 - a. Opal SSC 2.00 and 2.01 Feature
 - i. Feature Code = 0x0203
 - ii. Number of COM IDs \geq 1
 - iii. Number of Locking SP Admin Authorities \geq 4
 - iv. Number of Locking SP User Authorities \geq 8
 - b. Geometry Reporting Feature
 - i. Feature Code = 0x0003
 - c. Additional DataStore Table Feature
 - i. Feature Code = 0x0202
 - ii. Maximum number of DataStore Tables \geq 1
 - iii. Maximum total size of DataStore Tables \geq 0xA0000
 - iv. DataStore Table size alignment \geq 1

4.4 Properties

Notes

Begin Informative Text

The values in the Level 0 Discovery response reported in this section are examples and vary between implementations and locking states of ranges.

End Informative Text

Prerequisites

None

Test Sequence

- 1) Invoke Properties method with the following HostProperties values:
 - a. MaxComPacketSize = 4096 bytes
 - b. MaxPacketSize = 4076 bytes
 - c. MaxIndTokenSize = 4040 bytes

Expected Response

- 1) Step #1 SUCCEEDS
- 2) The SD returns the following values for TPer Properties:
 - a. MaxComPacketSize \geq 2048 bytes
 - b. MaxResponseComPacketSize \geq 2048 bytes
 - c. MaxPacketSize \geq 2028 bytes
 - d. MaxIndTokenSize \geq 1992 bytes
 - e. MaxPackets \geq 1
 - f. MaxSubpackets \geq 1
 - g. MaxMethods \geq 1
 - h. MaxSessions \geq 1
 - i. MaxAuthentications \geq 2
 - j. MaxTransactionLimit \geq 1
 - k. DefSessionTimeout \geq 0
- 3) The SD returns the following values for Host Properties:
 - a. MaxComPacketSize \geq 2048 bytes and \leq 4096 bytes
 - b. MaxPacketSize \geq 2028 bytes and \leq 4076 bytes
 - c. MaxIndTokenSize \geq 1992 bytes and \leq 4040 bytes

4.5 Taking Ownership of an SD

Notes

Begin Informative Content

The following test is to establish that an SD can be controlled by host software. Taking ownership is a key step in managing an SD.

End Informative Content

Prerequisites

None

Test Sequence

- 1) If Opal SSC 1.00, or if Opal SSC 2.00 or 2.01 and the Initial C_PIN_SID PIN Indicator value = 0, then
 - a. Invoke StartSession method with SPID = Admin SP UID
 - b. Invoke Get method to retrieve MSID's PIN column value from the C_PIN table
 - c. CLOSE_SESSION
 - d. Invoke StartSession method with SPID = Admin SP UID and HostSigningAuthority = SID authority UID.
 - e. SET_PASSWORD_FOR SID to < SID_PASSWORD>
 - f. CLOSE_SESSION
- 2) If Opal SSC 2.00 or 2.01, and the Initial C_PIN_SID PIN Indicator value <> 0, then obtain SID VU PIN value from the SD vendor
 - a. Invoke StartSession method with SPID = Admin SP UID, HostSigningAuthority = SID authority UID, and HostChallenge = SID C_PIN object's VU PIN column value
 - b. SET_PASSWORD_FOR SID to < SID_PASSWORD>
 - c. CLOSE_SESSION
- 3) If Opal SSC 2.00 or 2.01
 - a. Invoke StartSession method with SPID = Admin SP UID, HostSigningAuthority = SID authority UID, and HostChallenge = < SID_PASSWORD>
 - b. SET_PASSWORD_FOR Admin1 to < AdminSP_Admin1_ PASSWORD>
 - c. ENABLE Admin1
 - d. CLOSE_SESSION

Expected Response

- 1) If Opal SSC 1.00, or if Opal SSC 2.00 or 2.01 and the Initial C_PIN_SID PIN Indicator value = 0, then step #1 SUCCEEDS
- 2) If Opal SSC 2.00 or 2.01, and the Initial C_PIN_SID PIN Indicator value <> 0, then step #2 SUCCEEDS
- 3) If Opal SSC 2.00 or 2.01 then step #3 SUCCEEDS

4.6 Activate Locking SP when in Manufactured-Inactive State

Notes

Begin Informative Content

None

End Informative Content

Prerequisites

None

Test Sequence

- 1) Invoke StartSession method with SPID = Admin SP UID and HostSigningAuthority = SID authority UID
- 2) Invoke Activate method on Locking SP object
- 3) CLOSE_SESSION
- 4) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
- 5) CLOSE_SESSION

Expected Response

- 1) Steps #1-5 SUCCEED

4.7 Configuring Authorities

Notes

Begin Informative Content

The following sections describe the sequences of steps for setting the PIN Credential value for one or more Admin authorities, and enabling and setting the PIN Credential value for multiple User authorities.

End Informative Content

Prerequisites

None

Test Sequence – Opal SSC 1.00

- 1) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
- 2) SET_PASSWORD_FOR Admin1 to <Admin1_PASSWORD>
- 3) ENABLE User1
- 4) SET_PASSWORD_FOR User1 to <User1_PASSWORD>
- 5) ENABLE User4
- 6) SET_PASSWORD_FOR User4 to <User4_PASSWORD>
- 7) CLOSE_SESSION
- 8) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID

- 9) CLOSE_SESSION
- 10) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = User1 authority UID
- 11) CLOSE_SESSION
- 12) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = User4 authority UID
- 13) CLOSE_SESSION

Expected Response – Opal SSC 1.00

- 1) Steps #1-13 SUCCEED

Test Sequence – Opal SSC 2.00 and 2.01

- 1) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
- 2) SET_PASSWORD_FOR Admin1 to <Admin1_PASSWORD>
- 3) ENABLE Admin4
- 4) SET_PASSWORD_FOR Admin4 to <Admin4_PASSWORD>
- 5) ENABLE User1
- 6) SET_PASSWORD_FOR User1 to <User1_PASSWORD>
- 7) ENABLE User8
- 8) SET_PASSWORD_FOR User8 to <User8_PASSWORD>
- 9) CLOSE_SESSION
- 10) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
- 11) CLOSE_SESSION
- 12) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin4 authority UID
- 13) CLOSE_SESSION
- 14) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = User1 authority UID
- 15) CLOSE_SESSION
- 16) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = User8 authority UID
- 17) CLOSE_SESSION

Expected Response – Opal SSC 2.00 and 2.01

- 1) Steps #1-17 SUCCEED

4.8 Configuring Locking Objects (Locking Ranges)

Notes

Begin Informative Content

None

End Informative Content

Prerequisites

None

Test Sequence – Opal SSC 1.00

- 1) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
- 2) Invoke Set method on Locking_Range4. Configure the locking range as follows:
 - a) RangeStart = 0
 - b) RangeLength = 64
 - c) ReadLockEnabled = TRUE
 - d) WriteLockEnabled = TRUE
 - e) ReadLocked = FALSE
 - f) WriteLocked = FALSE
- 3) Invoke Set method on the BooleanExpr column of the ACE_Locking_Range4_Set_RdLocked ACE object to include the UIDs of the User1 and User4 Authority objects
- 4) Invoke Set method on the BooleanExpr column of the ACE_Locking_Range4_Set_WrLocked ACE object to include the UIDs of the User1 and User4 Authority objects
- 5) CLOSE_SESSION
- 6) Write the MAGIC_PATTERN over the entire Locking_Range4
- 7) Read over the entire Locking_Range4
- 8) Power cycle the SD
- 9) Read the entire Locking_Range4
- 10) Write the MAGIC_PATTERN over the entire Locking_Range4

Expected Response – Opal SSC 1.00

- 1) Steps #1-8 SUCCEED
- 2) The value returned from the Read command in step #7 is the MAGIC_PATTERN
- 3) Steps #9-10 return Data Protection Error

Test Sequence – Opal SSC 2.00 and 2.01

- 1) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
- 2) Invoke Set method on Locking_Range8. Configure the locking range as follows:
 - a) RangeStart = 0

- b) RangeLength = 64
 - c) ReadLockEnabled = TRUE
 - d) WriteLockEnabled = TRUE
 - e) ReadLocked = FALSE
 - f) WriteLocked = FALSE
 - g) LockOnReset = {0}
 - h) Adjust RangeStart and RangeLength according to the RangeAlignment
- 3) Invoke Set method on the BooleanExpr column of the ACE_Locking_Range8_Set_RdLocked ACE object to include the UIDs of the User1 and User8 Authority objects
 - 4) Invoke Set method on the BooleanExpr column of the ACE_Locking_Range8_Set_WrLocked ACE object to include the UIDs of the User1 and User8 Authority objects
 - 5) CLOSE_SESSION
 - 6) Write the MAGIC_PATTERN over the entire Locking_Range8
 - 7) Read the entire Locking_Range8
 - 8) Power cycle the SD
 - 9) Read the entire Locking_Range8
 - 10) Write the MAGIC_PATTERN over the entire Locking_Range8

Expected Response – Opal SSC 2.00 and 2.01

- 1) Steps #1-8 SUCCEED
- 2) The value returned from the Read command in step #7 is the MAGIC_PATTERN
- 3) Steps #9-10 return Data Protection Error

4.9 Unlocking Ranges

Notes

Begin Informative Content

None

End Informative Content

Prerequisites

None

Test Sequence – Opal SSC 1.00

- 1) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = User1 authority UID
- 2) Invoke Set method on the ReadLocked and WriteLocked columns of the Locking_Range4 Locking object with a value of FALSE
- 3) CLOSE_SESSION
- 4) Read the entire Locking_Range4

Expected Results – Opal SSC 1.00

- 1) Steps #1-4 SUCCEED

Test Sequence – Opal SSC 2.00 and 2.01

- 1) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = User1 authority UID
- 2) Invoke Set method on the ReadLocked and WriteLocked columns of the Locking_Range4 Locking object with a value of FALSE
- 3) CLOSE_SESSION
- 4) Read the entire Locking_Range8

Expected Results – Opal SSC 2.00 and 2.01

- 1) Steps #1-4 SUCCEED

4.10 Erasing Ranges

Notes

Begin Informative Content

None

End Informative Content

Prerequisites

None

Test Sequence – Opal SSC 1.00

- 1) Write the MAGIC_PATTERN over the entire Locking_Range4
- 2) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
- 3) Invoke Get method on Locking_Range4 to retrieve the ActiveKey column's value
- 4) Invoke GenKey method on the UID retrieved from Locking_Range4's ActiveKey column
- 5) CLOSE_SESSION
- 6) Attempt to read the entire Locking_Range4

Expected Response – Opal SSC 1.00

- 1) Steps #1-5 SUCCEED
- 2) The Read command in step #6 returns data that does not match the MAGIC_PATTERN

Test Sequence – Opal SSC 2.00 and 2.01

- 1) Write the MAGIC_PATTERN over the entire Locking_Range8
- 2) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
- 3) Invoke Get method on Locking_Range8 to retrieve the ActiveKey column's value
- 4) Invoke GenKey method on the UID retrieved from Locking_Range8's ActiveKey column
- 5) CLOSE_SESSION
- 6) Attempt to read the entire Locking_Range8

Expected Response – Opal SSC 2.00 and 2.01

- 1) Steps #1-5 SUCCEED
- 2) The Read command in step #6 returns data that does not match the MAGIC_PATTERN

4.11 Using the DataStore Table

Notes

Begin Informative Content

None

End Informative Content

Prerequisites

None

Test Sequence

- 1) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
- 2) Invoke Set method on the BooleanExpr column of the ACE_DataStore_Set_All ACE object to include the UID of the User1 Authority object
- 3) Invoke Set method on the BooleanExpr column of the ACE_DataStore_Get_All ACE object to include the UID of the User1 Authority object
- 4) CLOSE_SESSION
- 5) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = User1 authority UID
- 6) Invoke Set method to write the entire DataStore table with the MAGIC_PATTERN
- 7) CLOSE_SESSION
- 8) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = User1 authority UID
- 9) Invoke Get method on the DataStore Table to read the data of the DataStore Table
- 10) CLOSE_SESSION

Expected Response

- 1) Steps #1-10 SUCCEED
- 2) The Get method in step #9 returns the MAGIC_PATTERN

4.12 Enable MBR Shadowing**Notes**

Begin Informative Content

None.

End Informative Content

Prerequisites

None

Test Sequence

- 1) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
- 2) For Opal SSC 1.00 Invoke Set method on the BooleanExpr column of the ACE_MBRControl_Set_Done ACE object to include the UIDs of the User1 and User4 Authority objects
- 3) For Opal SSC 2.00 or 2.01 Invoke Set method on the BooleanExpr column of the ACE_MBRControl_Set_DoneToDOR ACE object to include the UIDs of the User1 and User4 Authority objects
- 4) Invoke Get method on the Rows column of the MBR Table Descriptor Object
- 5) Set Locking_Range4 RangeLength column to the LBA size of the MBR Table Descriptor Object + 10 LBAs

Note: Calculate the LBA size by dividing the value obtained in step 4 by the size of the LBA in bytes obtained through the discovery mechanisms of the underlying interface protocol.
- 6) Write 0s over the entire Locking_Range4
- 7) Invoke Set method to write the entire MBR table with the MAGIC_PATTERN
- 8) Invoke Set method on the Enable column of the MBRControl table with a value of TRUE
- 9) CLOSE_SESSION
- 10) Power cycle the SD
- 11) Write the MAGIC_PATTERN over the entire Locking_Range4
- 12) Read from LBA 0 to the size of the MBR Table
- 13) Read 10 LBAs starting immediately following the end of the MBR

Expected Response

- 1) Steps #1-10 SUCCEED

- 2) Step #11 returns Data Protection Error
- 3) The value returned from the Read command in step #12 matches the MAGIC_PATTERN
- 4) The value returned from the Read command in step #13 = 0s

4.13 MBR Done

Notes

Begin Informative Content

None

End Informative Content

Prerequisites

None

Test Sequence

- 1) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = User4 authority UID
- 2) Invoke Set method on the ReadLocked and WriteLocked columns of the Locking_Range4 Locking object with a value of FALSE
- 3) Invoke Set method on the Done column of the MBRControl table with a value of TRUE
- 4) CLOSE_SESSION
- 5) Read the entire Locking_Range4

Expected Response

- 1) Steps #1-5 SUCCEED
- 2) The value returned from the Read command in step #5 = 0s

4.14 Revert the Locking SP using SID, with Locking SP in Manufactured state

Notes

Begin Informative Content

None

End Informative Content

Prerequisites

None

Test Sequence

- 1) Write the MAGIC_PATTERN over the first 64 logical blocks of Locking_GlobalRange
- 2) Invoke StartSession method with SPID = Admin SP UID and HostSigningAuthority = SID authority UID
- 3) Invoke Revert method on Locking SP object
- 4) CLOSE_SESSION
- 5) Invoke StartSession method with SPID = Locking SP UID
- 6) Read the first 64 logical blocks of Locking_GlobalRange

Expected Response

- 1) Steps #1-4 SUCCEED
- 2) The StartSession method in step #5 results in a SyncSession method with a status code of INVALID_PARAMETER
- 3) The Read command in step #6 returns data that does not match the MAGIC_PATTERN

4.15 Revert the Admin SP using SID, with Locking SP in Manufactured-Inactive state

Notes

Begin Informative Content

None

End Informative Content

Prerequisites

None

Test Sequence

- 1) Write the MAGIC_PATTERN over the first 64 logical blocks, beginning at LBA0
- 2) Invoke StartSession method with SPID = Admin SP UID and HostSigningAuthority = SID authority UID
- 3) Invoke Revert method on Admin SP object
- 4) If the “Behavior of C_PIN_SID Pin upon TPer Revert” from the return of Level 0 Discovery = 0 then
 - a. Invoke StartSession method with SPID = Admin SP UID, HostSigningAuthority = SID authority UID, and HostChallenge = C_PIN_MSID PIN column valueElse
 - a. Invoke StartSession method with SPID = Admin SP UID, HostSigningAuthority = SID authority UID, and HostChallenge = C_PIN_SID VU PIN column value
- 5) CLOSE_SESSION
- 6) Invoke StartSession method with SPID = Locking SP

- 7) Read the first 64 logical blocks, beginning at LBA0

Expected Response

- 1) Steps #1-5 SUCCEED
- 2) The StartSession method in step #6 results in a SyncSession method with a status code of INVALID_PARAMETER
- 3) The Read command in step #7 returns data that matches the MAGIC_PATTERN

4.16 Revert the Admin SP using SID, with Locking SP in Manufactured state

Notes

Begin Informative Content

None

End Informative Content

Prerequisites

- 1) Locking SP is in the Manufactured state
- 2) SID's PIN column value is set to < SID_PASSWORD> value in the SID's C_PIN credential PIN column
- 3) Determining support for the Revert feature:
 - a. Invoke StartSession method with SPID = Admin SP UID
 - b. Invoke Get method on UID 00 00 00 06 00 00 02 02 to determine support

Test Sequence

- 1) Write the MAGIC_PATTERN over the first 64 logical blocks, beginning with LBA0
- 2) Invoke StartSession method with SPID = Admin SP UID and HostSigningAuthority = SID authority UID
- 3) Invoke Revert method on Admin SP object
- 4) If the "Behavior of C_PIN_SID Pin upon TPer Revert" from the return of Level 0 Discovery = 0 then
 - a. Invoke StartSession method with SPID = Admin SP UID, HostSigningAuthority = SID authority UID, and HostChallenge = C_PIN_MSID PIN column value
 Else
 - a. Invoke StartSession method with SPID = Admin SP UID, HostSigningAuthority = SID authority UID, and HostChallenge = C_PIN_SID VU PIN column value
- 5) CLOSE_SESSION
- 6) Invoke StartSession method with SPID = Locking SP UID
- 7) Read the first 64 logical blocks, beginning at LBA0

Expected Response

- 1) Steps #1-5 SUCCEED
- 2) The StartSession method in step #6 results in a SyncSession method with a status code of INVALID_PARAMETER
- 3) The Read command in step #7 returns data that does not match the MAGIC_PATTERN

4.17 Revert Admin SP using Admin1, with Locking SP in Manufactured state – Opal SSC 2.00 and 2.01

Notes

Begin Informative Content

See [2] [3] [4] for support requirements.

End Informative Content

Prerequisites

- 1) Locking SP is in the Manufactured state
- 2) Admin1 authority is enabled
- 3) Admin1's PIN column value is set to < Admin1_PASSWORD> value in the Admin1's C_PIN credential PIN column

Test Sequence

- 1) Write the MAGIC_PATTERN over the first 64 logical blocks, beginning with LBA0
- 2) Invoke StartSession method with SPID = Admin SP UID and HostSigningAuthority = Admin1 authority UID
- 3) Invoke Revert method on Admin SP object
- 4) If the "Behavior of C_PIN_SID Pin upon TPer Revert" from the return of Level 0 Discovery = 0 then
 - a. Invoke StartSession method with SPID = Admin SP UID, HostSigningAuthority = SID authority UID, and HostChallenge = C_PIN_MSID PIN column value
 Else
 - a. Invoke StartSession method with SPID = Admin SP UID, HostSigningAuthority = SID authority UID, and HostChallenge = C_PIN_SID VU PIN column value
- 5) CLOSE_SESSION
- 6) Invoke StartSession method with SPID = Locking SP UID
- 7) Read the first 64 logical blocks beginning with LBA0

Expected Response

- 1) Steps #1-5 SUCCEED

- 2) The StartSession method in step #6 results in a SyncSession method with a status code of INVALID_PARAMETER
- 3) The Read command in step #7 returns data that does not match the MAGIC_PATTERN

4.18 Revert Admin SP using PSID, with Locking SP in Manufactured state

Notes

Begin Informative Content

None

End Informative Content

Prerequisites

- 1) Locking SP is in the Manufactured state
- 2) PSID Feature Set is implemented
- 3) PSID value can be obtained

Test Sequence

- 1) Write the MAGIC_PATTERN over the first 64 logical blocks beginning with LBA0
- 2) Invoke StartSession method with SPID = Admin SP UID, HostSigningAuthority = PSID authority UID, and HostChallenge = PSID authority's credential obtained from the VU PSID delivery mechanism
- 3) Invoke Revert method on Admin SP object
- 4) If the "Behavior of C_PIN_SID Pin upon TPer Revert" from the return of Level 0 Discovery = 0 then
 - a. Invoke StartSession method with SPID = Admin SP UID, HostSigningAuthority = SID authority UID, and HostChallenge = C_PIN_MSID PIN column valueElse
 - a. Invoke StartSession method with SPID = Admin SP UID, HostSigningAuthority = SID authority UID, and HostChallenge = C_PIN_SID VU PIN column value
- 5) CLOSE_SESSION
- 6) Invoke StartSession method with SPID = Locking SP UID
- 7) Read the first 64 logical blocks beginning with LBA0

Expected Response

- 1) Steps #1-5 SUCCEED
- 2) The StartSession method in step #6 results in a SyncSession method with a status code of INVALID_PARAMETER
- 3) The Read command in step #7 returns data that does not match the MAGIC_PATTERN

5. Specific Functionality

These test cases reflect specific functionality that SHALL be performed on a device that complies with the Opal SSC 1.00, 2.00 or 2.01 specifications. Unless otherwise specified within a test case, the expected result of each step is that the step SHALL SUCCEED.

5.1 Common Prerequisites

Unless otherwise noted, the following set of prerequisites apply for each test in this section:

1. SD is in Awaiting IF-SEND
2. Locking SP is in Manufactured state
3. The values of any credentials used are known
4. All StartSession method HostChallenge parameters use the current C_PIN object's PIN column value for the Authority used in the HostSigningAuthority parameter
5. All sessions are Read-Write sessions
6. No open sessions exist at the start of the Test Sequence

5.2 Transaction

Notes

Begin Informative Content

There are two tests performed relating to Transactions:

1. Test 1 attempts to write the entire MBR Table with the MAGIC_PATTERN.
2. Test 2 attempts to write the entire MBR Table with 0s, and then close the session without committing the Transaction.

Since Session Timeout is VU, test results may be NA if session timeout occurs or if the transaction cannot be committed.

End Informative Content

Prerequisites – Test 1

- 1) Knowledge of the MBR Table size
- 2) For Opal SSC 2.00 and 2.01 knowledge of the MandatoryWriteGranularity Column value for the MBR Table

Test Sequence – Test 1

- 1) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
- 2) For Opal SSC 1.00
 - a. Invoke Set method to write the entire MBR Table with 0s
- 3) For Opal SSC 2.00 and 2.01

- a. Invoke Set method to write the entire MBR Table with 0s while adhering to the MandatoryWriteGranularity requirements
- 4) CLOSE_SESSION if the write is successful, or if the session aborts due to a timeout, exit the test and record result as NA
- 5) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
- 6) Send a subpacket that contains a StartTransaction token with a status code of 0x00
- 7) For Opal SSC 1.00
 - a. Invoke Set method to write the entire MBR Table with the MAGIC_PATTERN
- 8) For Opal SSC 2.00 and 2.01
 - a. Invoke Set method to write the entire MBR Table with the MAGIC_PATTERN while adhering to the MandatoryWriteGranularity requirements
- 9) Send a subpacket that contains an End Transaction token with a status code of 0x00
- 10) CLOSE_SESSION if the SD responds with an End Transaction token with a status code of 0x00, or if the session aborts due to a timeout exit the test and record result as NA
- 11) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
- 12) Invoke Get method on the MBR Table to read the data of the MBR Table
- 13) CLOSE_SESSION

Expected Response – Test 1

- 1) Steps #1-13 SUCCEED
- 2) The Get method in step #12 returns the MAGIC_PATTERN
- 3) If the session is aborted on step #4 or step #10, the result of this test is “NA (Not Applicable)”

Prerequisites – Test 2

- 1) Steps #1-13 in Test 1 SUCCEED

Test Sequence – Test 2

- 1) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
- 2) Send a subpacket that contains a StartTransaction token with a status code of 0x00
- 3) For Opal SSC 1.00
 - a) Invoke Set method to write the entire MBR Table with 0s
- 4) For Opal SSC 2.00 and 2.01
 - a) Invoke Set method to write the entire MBR Table with 0s while adhering to the MandatoryWriteGranularity requirements
- 5) CLOSE_SESSION if the write is successful, or if the session aborts due to a timeout exit the test and record result as NA

- 6) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
- 7) Invoke Get method on the MBR Table to read the data of the MBR Table
- 8) CLOSE_SESSION

Expected Response – Test 2

- 1) Steps #1-8 SUCCEED
- 2) The Get method in step #7 returns the MAGIC_PATTERN. If the session is aborted on step #5, the result of this test is “NA (Not Applicable)”

5.3 IF-RECV Behavior Tests

5.3.1 TPer Response to IF-RECV when in Awaiting IF-SEND

Notes

Begin Informative Content

None

End Informative Content

Prerequisites

- 1) In Awaiting IF-SEND

Test Sequence

- 1) Issue an IF-RECV command

Expected Response

- 1) Steps #1 SUCCEEDS
- 2) IF-RECV in step #1 has a ComPacket header value of “All Response(s) returned - no further data”, (See [1])

5.3.2 TPer Response to IF-RECV with Insufficient Transfer Length

Notes

Begin Informative Content

None

End Informative Content

Prerequisites

None

Test Sequence

- 1) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
- 2) Invoke Get method on the DataStore Table to retrieve 1024 Rows. For the IF-RECV command issued by the Host to retrieve the result, the IF-RECV command has a transfer length of 1
- 3) Issue IF-RECV command to retrieve the result with the transfer length based on the MinTransfer value in the IF-RECV response to step #2
- 4) CLOSE_SESSION

Expected Response

- 1) Step #1-4 SUCCEED
- 2) IF-RECV in step #2 has a ComPacket header value of "Response ready, insufficient transfer length request", see [1]

5.4 TryLimit

Notes

Begin Informative Content

None

End Informative Content

Prerequisites

- 1) User1 is enabled

Test Sequence

- 1) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
- 2) Invoke Get method on Admin1's C_PIN Object to retrieve the TryLimit Column's value
- 3) Invoke Get method on User1's C_PIN Object to retrieve the TryLimit Column's value
- 4) CLOSE_SESSION
- 5) Invoke StartSession method with SPID = Admin SP UID and HostSigningAuthority = SID authority UID
- 6) Invoke Get method on SID's C_PIN Object to retrieve the TryLimit Column's value
- 7) CLOSE_SESSION
- 8) If SID C_PIN Object has a TryLimit Column value >0, then

- a) Invoke StartSession method with SPID = Admin SP UID, HostSigningAuthority = SID authority UID, and HostChallenge = a value that does not match the current SID C_PIN object's PIN column value, until SID C_PIN object's Tries value = SID C_PIN object's TryLimit value
 - b) Invoke StartSession method with SPID = Admin SP UID and HostSigningAuthority = SID authority UID
- 9) If Admin1 C_PIN Object has a TryLimit Column value >0, then
- a) Invoke StartSession method with SPID = Locking SP UID, HostSigningAuthority = Admin1 authority UID, and HostChallenge = a value that does not match the current Admin1 C_PIN object's PIN column value, until Admin1 C_PIN object's Tries value = Admin1 C_PIN object's TryLimit value
 - b) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
- 10) If User1 C_PIN Object has a TryLimit Column value >0, then
- a) Invoke StartSession method with SPID = Locking SP UID, HostSigningAuthority = User1 authority UID, and HostChallenge = a value that does not match the current User1 C_PIN object's PIN column value, until User1 C_PIN object's Tries value = User1 C_PIN object's TryLimit value
 - b) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = User1 authority UID

Expected Response

- 1) Steps #1-7 SUCCEED
- 2) Steps #8-10 FAIL for any Authority with a TryLimit value >0.
- 3) StartSession method with the correct C_PIN column value in steps #8-10 results in SyncSession method with a status code of NOT_AUTHORIZED or AUTHORITY_LOCKED_OUT

5.5 Tries Reset

Notes

Begin Informative Content

The following test verifies that the value of Tries is reset upon successful authentication.

End Informative Content

Prerequisites

- 1) User1 is enabled

Test Sequence

- 1) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
- 2) Invoke Get method on Admin1's C_PIN Object to retrieve the TryLimit Column's value
- 3) Invoke Get method on User1's C_PIN Object to retrieve the TryLimit Column's value
- 4) CLOSE_SESSION

- 5) Invoke StartSession method with SPID = Admin SP UID and HostSigningAuthority = SID authority UID.
- 6) Invoke Get method on SID's C_PIN Object to retrieve the TryLimit Column's value
- 7) CLOSE_SESSION
- 8) If SID C_PIN Object has a TryLimit Column value > 1, then
 - a) Invoke StartSession method with SPID = Admin SP UID, HostSigningAuthority = SID authority UID, and HostChallenge = a value that does not match the current SID C_PIN object's PIN column value, until SID C_PIN object's Tries value = SID C_PIN object's TryLimit value -1
 - b) Invoke StartSession method with SPID = Admin SP UID and HostSigningAuthority = SID authority UID.
 - c) Invoke Get method on the Tries Column of the SID Authority's C_PIN Object
 - d) CLOSE_SESSION
- 9) If User1 C_PIN Object has a TryLimit Column value >1, then
 - a) Invoke StartSession method with SPID = Locking SP UID, HostSigningAuthority = User1 authority UID, and HostChallenge = a value that does not match the current User1 C_PIN object's PIN column value, until User1 C_PIN object's Tries value = User1 C_PIN object's TryLimit value -1
 - b) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = User1 authority UID.
 - c) CLOSE_SESSION
 - d) Invoke Get method on the Tries Column of the User1 Authority's C_PIN Object
 - e) CLOSE_SESSION
- 10) If Admin1 C_PIN Object has a TryLimit Column value > 1, then
 - a) Invoke StartSession method with SPID = Locking SP UID, HostSigningAuthority = Admin1 authority UID, and HostChallenge = a value that does not match the current Admin1 C_PIN object's PIN column value, until Admin1 C_PIN object's Tries value = Admin1 C_PIN object's TryLimit value -1
 - b) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
 - c) Invoke Get method on the Tries Column of the Admin1 Authority's C_PIN Object
 - d) CLOSE_SESSION

Expected Response

- 1) Steps #1-10 SUCCEED
- 2) For each Authority with a TryLimit column value > 1, that Authority's C_PIN Tries column value = 0

5.6 Tries Reset on Power Cycle

Notes

Begin Informative Content

The following test verifies that the value of Tries is reset upon power cycle.

End Informative Content**Prerequisites**

- 1) User1 is enabled

Test Sequence

- 1) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
- 2) Invoke Get method on Admin1's C_PIN Object to retrieve the TryLimit Column's value
- 3) Invoke Get method on User1's C_PIN Object to retrieve the TryLimit Column's value
- 4) CLOSE_SESSION
- 5) Invoke StartSession method with SPID = Admin SP UID and HostSigningAuthority = SID authority UID.
- 6) Invoke Get method on SID's C_PIN Object to retrieve the TryLimit Column's value
- 7) CLOSE_SESSION
- 8) If SID C_PIN Object has a TryLimit Column value >0, then
 - a) Invoke StartSession method with SPID = Admin SP UID, HostSigningAuthority = SID authority UID, and HostChallenge = a value that does not match the current SID C_PIN object's PIN column value, until SID C_PIN object's Tries value = SID C_PIN object's TryLimit value.
- 9) If Admin1 C_PIN Object has a TryLimit Column value >0, then
 - a) Invoke StartSession method with SPID = Locking SP UID, HostSigningAuthority = Admin1 authority UID, and HostChallenge = a value that does not match the current Admin1 C_PIN object's PIN column value, until Admin1 C_PIN object's Tries value = Admin1 C_PIN object's TryLimit value.
- 10) If User1 C_PIN Object has a TryLimit Column value >0, then
 - a) Invoke StartSession method with SPID = Locking SP UID, HostSigningAuthority = User1 authority UID, and HostChallenge = a value that does not match the current User1 C_PIN object's PIN column value, until User1 C_PIN object's Tries value = User1 C_PIN object's TryLimit value.
- 11) Power cycle the SD
- 12) If SID C_PIN Object has a TryLimit Column value >0, then
 - a) Invoke StartSession method with SPID = Admin SP UID and HostSigningAuthority = SID authority UID
 - b) Invoke Get method on SID Authority's C_PIN Tries Column
 - c) CLOSE_SESSION
- 13) If Admin1 C_PIN Object has a TryLimit Column value >0, then
 - a) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
 - b) Invoke Get method on Admin1 Authority's C_PIN Tries Column
 - c) CLOSE_SESSION
- 14) If User1 C_PIN Object has a TryLimit Column value >0, then

- a) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = User1 authority UID
- b) Invoke Get method on User1 Authority's C_PIN Tries Column
- c) CLOSE_SESSION

Expected Response

- 1) Steps #1-14 SUCCEED
- 2) Each Authority's C_PIN Tries column value = 0, or
 - a) If SID C_PIN TryLimit Column value > 0, then
 - i) Admin SP session opens successfully
 - ii) Get method on SID Authority's C_PIN Tries Column returns 0
 - b) If Admin1 C_PIN TryLimit Column value > 0, then
 - i) Locking SP session opens successfully
 - ii) Get method on Admin1 Authority's C_PIN Tries Column returns 0
 - c) If User1 C_PIN TryLimit Column value > 0, then
 - i) Locking SP session opens successfully
 - ii) Get method on User1 Authority's C_PIN Tries Column returns 0

5.7 Next

Notes

Begin Informative Content

Testing of Next method is to verify the existence of a single UID.

End Informative Content

Prerequisites

None

Test Sequence

- 1) Invoke StartSession method with SPID = Locking SP UID
- 2) Invoke Get method on the LockingInfo Table's MaxRanges Column
- 3) Invoke Next method on the Locking Table with an empty parameter list
- 4) Invoke Next method on the Locking Table with the Where parameter set to the UID of Locking_Range1 and the Count parameter set to 1
- 5) CLOSE_SESSION

Expected Response

- 1) Steps #1-5 SUCCEED
- 2) Step #3
 - a) returns a list of UIDs where the number of values = the MaxRanges value + 1, and
 - b) the first four bytes of each UID returned are 0x00000802
- 3) Step #4 returns a list that contains only the UID of Locking_Range1

5.8 Host Session Number (HSN)

Notes

Begin Informative Content

Test the Host Session Number to verify that the SD responses with the corresponding Host Session Number provided by the host.

End Informative Content

Prerequisites

None

Test Sequence

- 1) Invoke StartSession method with HostSessionID = ARBITRARILY_VARYING HSN, SPID = Admin SP UID, and HostSigningAuthority = Admin1 authority UID
- 2) Invoke Get method on MSID C_PIN credential's PIN Column
- 3) CLOSE_SESSION

Expected Response

- 1) Steps #1-3 SUCCEED
- 2) The StartSession method in step #1 results in a SyncSession method with the same HSN as parameterized in the StartSession method
- 3) The Packet received in step #2 that contains the Get method response has the same HSN as parameterized in the StartSession method

5.9 RevertSP on Locking SP KeepGlobalRangeKey Parameter Effects

Notes

Begin Informative Content

See [2] for support requirements on RevertSP and KeepGlobalRangeKey. There are three tests in this test case. Each must be performed.

End Informative Content

Prerequisites – Test 1

None

Test Sequence – Test 1

- 1) Write the MAGIC_PATTERN over the first 64 logical blocks of Locking_GlobalRange
- 2) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
- 3) Invoke RevertSP method with the KeepGlobalRangeKey omitted
- 4) Read the first 64 logical blocks of Locking_GlobalRange

Expected Response – Test 1

- 1) Steps #1-3 SUCCEED
- 2) The Read command in step #4 returns data that does not match the MAGIC_PATTERN

Prerequisites – Test 2

None

Test Sequence – Test 2

- 1) Write the MAGIC_PATTERN over the first 64 logical blocks of Locking_GlobalRange
- 2) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
- 3) Invoke RevertSP method with the KeepGlobalRangeKey present and set to FALSE
- 4) Read the first 64 logical blocks of Locking_GlobalRange

Expected Response – Test 2

- 1) Steps #1-3 SUCCEED
- 2) The Read command in step #4 returns data that does not match the MAGIC_PATTERN

Prerequisites – Test 3

None

Test Sequence – Test 3

- 1) Write the MAGIC_PATTERN over the first 64 logical blocks of Locking_GlobalRange
- 2) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
- 3) Invoke RevertSP method with the KeepGlobalRangeKey present and set to TRUE

- 4) Read the first 64 logical blocks of Locking_GlobalRange

Expected Response – Test 3

- 1) Steps #1-4 SUCCEED
- 2) The Read command in step #4 returns data that matches the MAGIC_PATTERN

5.10 Range Alignment Verification – Opal SSC 2.00 and 2.01

Notes

Begin Informative Content

None

End Informative Content

Prerequisites

- 1) Confirm the LockingInfo table AlignmentRequired column = TRUE. If AlignmentRequired = FALSE do not perform the test.

Test Sequence

- 1) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
- 2) Invoke Get method on the LockingInfo Table to retrieve the LogicalBlockSize, AlignmentGranularity and LowestAlignedLBA column values
- 3) If AlignmentGranularity is > 1, then
 - a) Invoke Set method on RangeLength and RangeStart columns with RangeStart and RangeLength values satisfying the conditions:
 - i) $[(\text{RangeStart} - \text{LowestAlignedLBA}) \% \text{AlignmentGranularity}] = 0$
 - ii) $[\text{RangeLength} \% \text{AlignmentGranularity}] = 0$
 - b) Invoke Set method on RangeLength and RangeStart columns with RangeStart and RangeLength values satisfying the following conditions:
 - i) Invoke Set method on the Locking Table with the chosen RangeStart and RangeLength values
- 4) CLOSE_SESSION

Expected Response

- 1) Steps #1-4 SUCCEED

5.11 Byte Table Access Granularity – Opal SSC 2.00 and 2.01

Notes

Begin Informative Content

None

End Informative Content

Prerequisites

- 1) Confirm the TPer requires specific Alignment granularity for byte tables. If Alignment granularity is not required by the TPer, do not perform the test.

Test Sequence

- 1) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
- 2) Invoke Get method on the DataStore object in the Table table to retrieve the MandatoryWriteGranularity column value
- 3) Invoke Set method to write the DataStore table with a number of 0s = a non-zero multiple of the MandatoryWriteGranularity column value
- 4) CLOSE_SESSION

Expected Response

- 1) Steps #1-4 SUCCEED

5.12 Stack Reset

Notes

Begin Informative Content

Reference SD vendor documentation to determine whether the command is supported.

End Informative Content

Prerequisites

- 1) Protocol 2 is supported
- 2) User1 is not enabled

Test Sequence

- 1) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID

- 2) Send a subpacket that contains a StartTransaction token with a status code of 0x00Invoke Set method on the Enabled Column of User1 Authority with a value of TRUE
- 3) Issue Protocol Stack Reset command
- 4) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
- 5) Invoke Get method to retrieve the value of the Enabled Column of User1 Authority
- 6) CLOSE_SESSION

Expected Response

- 1) Steps #1-6 SUCCEED
- 2) The Get method in step #5 returns a value of FALSE

5.13 TPer Reset – Opal SSC 2.00 and 2.01

Notes

Begin Informative Content

None

End Informative Content

Prerequisites

- 1) ProgrammaticResetEnable set to TRUE
- 2) Locking_Range8 has ReadLocked and WriteLocked columns set to FALSE
- 3) ReadLockEnabled and WriteLockEnabled columns are set to TRUE
- 4) LockOnReset column value includes Programmatic

Test Sequence

- 1) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID.
- 2) Issue the TPER_ RESET command
- 3) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID.
- 4) Invoke Get method on Locking_Range8's ReadLocked and WriteLocked columns
- 5) CLOSE_SESSION

Expected Response

- 1) Steps #1-5 SUCCEED
- 2) The Get method in step #4 returns values of TRUE

5.14 Authenticate – Opal SSC 2.0 and 2.01

Notes

Begin Informative Content

None

End Informative Content

Prerequisites

None

Test Sequence

- 1) Invoke StartSession method with SPID = Admin SP UID
- 2) Invoke Authenticate method with Authority = SID Authority UID and Proof = C_PIN_SID PIN column value
- 3) Invoke Get method on UID Column of SID C_PIN
- 4) CLOSE_SESSION

Expected Response

- 1) Steps #1-4 SUCCEED
- 2) The Get method in step #3 returns the C_PIN_SID PIN object's UID column value

5.15 Session Abort

Notes

Begin Informative Content

None

End Informative Content

Prerequisites

None

Test Sequence

- 1) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
- 2) Send a subpacket that contains a StartTransaction token with a status code of 0x00Change Admin1 PIN to a random value. Note: for Opal SSC 2.00 and 2.01 the Random method can be used to generate new PIN column value

- 3) CLOSE_SESSION
- 4) Invoke StartSession method with SPID = Locking SP UID, HostSigningAuthority = Admin1 authority UID, and HostChallenge = the original C_PIN_Admin1 PIN column value
- 5) CLOSE_SESSION

Expected Response

- 1) Steps #1-5 SUCCEED

5.16 Random

Notes

Begin Informative Content

This test is not intended to guarantee the quality of the RNG.

End Informative Content

Prerequisites

None

Test Sequence

- 1) Invoke StartSession method with SPID = Locking SP UID
- 2) Invoke Random method with a Count = 32
- 3) Invoke Random method with a Count = 32
- 4) CLOSE_SESSION

Expected Response

- 1) Steps #1-4 SUCCEED
- 2) The value returned by the Random method in step #2 is 32 bytes long and does not contain either all 0s or all 1s
- 3) The value returned from the Random method in step #3 is 32 bytes long and does not contain either all 0s or all 1s
- 4) The two values returned from the Random method in steps #2 and #3 are different

5.17 CommonName – Opal SSC 2.00 and 2.01

Notes

Begin Informative Content

None

End Informative Content

Prerequisites

- 1) Admin7 is enabled

Test Sequence

- 1) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
- 2) Invoke a 32 byte Set method on the CommonName column of the Admin7 authority object using the MAGIC_PATTERN
- 3) Invoke a 32 byte Set method on the CommonName column of Locking_Range7 using the MAGIC_PATTERN
- 4) Invoke Get method on the CommonName column of the Admin7 authority object
- 5) Invoke Get method on the CommonName column of Locking_Range 7
- 6) CLOSE_SESSION

Expected Response

- 1) Steps #1-6 SUCCEED
- 2) The values returned from the Get methods in steps #4-5 are the same as the values previously Set in steps #2-3

5.18 DataStore Table – Opal SSC 2.00 and 2.01

Notes

Begin Informative Content

Only one of the following tests is performed based on the value of the Maximum Number of DataStore Tables field in the DataStore Table Feature Descriptor.

End Informative Content

Prerequisites – Test 1

- 1) In the DataStore Table Feature Descriptor, the Maximum Number of DataStore Tables field value = 1
- 2) Locking SP is in the Manufacture-Inactive State

Test Sequence – Test 1

- 1) Issue Level 0 Discovery command to retrieve the DataStore Table Size Alignment field
- 2) Invoke StartSession method with SPID = Admin SP UID and HostSigningAuthority = SID authority UID

- 3) Invoke Activate method on the Locking SP with a DataStoreTableSize parameter value = 1 x the value of the DataStore Table Size Alignment field of the Level 0 Discovery Feature Descriptor
- 4) CLOSE_SESSION
- 5) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
- 6) Invoke Get method to retrieve the DataStore table's Rows column value from the Table table
- 7) CLOSE_SESSION

Expected Response – Test 1

- 1) Steps #1-7 SUCCEED
- 2) The Get method in step #6 returns a value = the DataStoreTableSize parameter value in step #3

Prerequisites – Test 2

- 1) In the DataStore Table Feature Descriptor, the Maximum Number of DataStore Tables field value > 1
- 2) Locking SP is in the Manufactured-Inactive State

Test Sequence – Test 2

- 1) Issue Level 0 Discovery command to retrieve the DataStore Table Size Alignment field
- 2) Invoke StartSession method with SPID = Admin SP UID and HostSigningAuthority = SID authority UID
- 3) Invoke Activate method with a DataStoreTableSize parameter value containing a number of items = the Maximum Number of DataStore Tables field, with values = the value of the DataStore Table Size Alignment field.
- 4) CLOSE_SESSION
- 5) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
- 6) Invoke Get method to retrieve each DataStore table's Rows column value from the Table table
- 7) CLOSE_SESSION

Expected Response – Test 2

- 1) Steps #1-5 SUCCEED
- 2) For each DataStore Table, the Get method in step #6 returns a value = the DataStoreTableSize parameter value in step #3

5.19 Range Crossing Behavior

Notes

Begin Informative Content

Test that the range crossing behavior is as specified by the returned value for range crossing.

Determine support for feature via Level 0 Discovery.

*End Informative Content***Prerequisites**

- 1) Locking_Range4 length is non-zero and does not span the entire SD
- 2) Locking_GlobalRange and Locking_Range4 are unlocked

Test Sequence

- 1) Issue a Write command with the MAGIC_PATTERN, with a beginning LBA in Locking_Range4 and ending LBA in Locking_GlobalRange
- 2) Issue a Read command, with a beginning LBA in Locking_Range4 and ending LBA in Locking_GlobalRange

Expected Response

- 1) If Range Crossing is supported, then steps #1-2 SUCCEED
- 2) If Range Crossing is not supported, then steps #1-2 FAIL. The Write command in step #1 and the Read command in step #2 return Other Invalid Command Parameter

6. Error Test Cases

The goal of this section is twofold: a) to reduce the overall number of error tests, and b) to require only a single instance of a common error test in the test cases. All possible unique error responses defined in the Opal SSC specifications are included in at least one test case.

Unless otherwise noted within a specific test case, session status is deemed to remain unaffected by the performance of any tests in this section.

This section does not include any tests where multiple errors are encoded in a payload from the host. Each test case only tests for a single error condition; however, some test cases may result in different possible error responses.

For every test case in this specification that specifies an error status code response, session abort SHALL be an acceptable response. In the case of session abort, the SD sending a CloseSession response SHALL be acceptable.

6.1 Common Prerequisites

Unless otherwise notes, the following set of prerequisites apply for each test in this section:

1. SD is in Awaiting IF-SEND
2. Locking SP is in Manufactured state
3. The values of any credentials used are known
4. All sessions are Read-Write sessions
5. No sessions are open

6.2 Native Protocol Read/Write Locked Error Responses

Notes

Begin Informative Content

None

End Informative Content

Prerequisites

- 1) Locking_Range4 ReadLockEnabled, WriteLockEnabled, ReadLocked and WriteLocked column values = TRUE

Test Sequence

- 1) Issue a Write command on Locking_Range4
- 2) Issue a Read command on Locking_Range4

Expected Response

- 1) Steps #1-2 FAIL

- 2) For the Write command in step #1 and the Read command in step #2, the SD SHALL:
 - a) Transfer no data
 - b) Return a Data Protection Error, (See [2])

6.3 General – IF-SEND/IF-RECV Synchronous Protocol

Notes

Begin Informative Content

None

End Informative Content

Prerequisites

None

Test Sequence

- 1) Invoke Properties method within an IF-SEND using a valid ComID and do not retrieve the response with an IF-RECV
- 2) Invoke Properties method using the ComID from the previous step

Expected Response

- 1) Step #1 SUCCEEDS
- 2) Step #2 FAILS. The IF-SEND command returns Synchronous Protocol Violation error

6.4 Invalid ComPacket Header Length Field

Notes

Begin Informative Content

None

End Informative Content

Prerequisites

None

Test Sequence

- 1) Invoke Properties method to determine SD's MaxComPacketSize

- 2) Invoke Properties method with a ComPacket Header's Length value larger than the SD's MaxComPacketSize

Expected Response

- 1) Steps #1-2 SUCCEED
- 2) IF-RECV in step #2 has a ComPacket header value of "All Response(s) returned - no further data", (See [1])

6.5 Invalid SessionID - Regular Session

Notes

Begin Informative Content

None

End Informative Content

Prerequisites

None

Test Sequence

- 1) Invoke StartSession method with SPID = Admin SP UID
- 2) Invoke Get method on MSID's credential object in C_PIN table with a Packet SessionID value <> the current SessionID value
- 3) CLOSE_SESSION

Expected Responses

- 1) Steps #1-3 SUCCEED
- 2) IF-RECV in step #2 has a ComPacket header value of "All Response(s) returned - no further data", (See [1])

6.6 Unexpected Token Outside of Method – Regular Session

Notes

Begin Informative Content

None

End Informative Content

Prerequisites

- 1) User1 authority object's Enabled Column is set to TRUE

Test Sequence

- 1) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
- 2) Invoke Set method on the Enabled Column of User1 Authority with a value of FALSE and EndList Token before the Call Token
- 3) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
- 4) Invoke Get method on the Enabled Column of User1 Authority
- 5) CLOSE_SESSION

Expected Response

- 1) Steps #1-5 SUCCEED
- 2) IF-RECV in step #2 has a ComPacket header value of "All Response(s) returned - no further data", (See [1])
- 3) Step #4 Get method on the Enabled Column of the User1 Authority returns FALSE

6.7 Unexpected Token in Method Header – Regular Session

Notes

Begin Informative Content

None

End Informative Content

Prerequisites

None

Test Sequence

- 1) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
- 2) Invoke Set method on the Enabled Column of User1 Authority with a value of FALSE and an EndList Token immediately after the Call Token
- 3) CLOSE_SESSION

Expected Response

- 1) Step #1 SUCCEEDS
- 2) Step #2 Set method returns NOT_AUTHORIZED
- 3) Step #3 SUCCEEDS

6.8 Unexpected Token Outside of Method – Control Session

Notes

Begin Informative Content

None

End Informative Content

Prerequisites

None

Test Sequence

- 1) Invoke StartSession method with SPID = Locking SP UID and an EndList Token before the Call Token
- 2) Invoke StartSession method with SPID = Locking SP UID

Expected Response

- 1) IF-RECV in step #1 has a ComPacket header value of “All Response(s) returned - no further data”, (See [1])
- 2) Steps #2 SUCCEEDS

6.9 Unexpected Token in the Method Parameter List – Control Session

Notes

Begin Informative Content

None

End Informative Content

Prerequisites

None

Test Sequence

- 1) Invoke Properties method with StartList immediately after the Parameter StartList

Expected Response

- 1) The Properties method in step #1 returns INVALID_PARAMETER

6.10 Exceeding Transaction Limit

Notes

Begin Informative Content

None

End Informative Content

Prerequisites

None

Test Sequence

- 1) Invoke Properties method to identify MaxTransactionLimit
- 2) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
- 3) Send a subpacket that contains MaxTransactionLimit + 1 StartTransaction Tokens
- 4) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
- 5) CLOSE_SESSION

Expected Response

- 1) Steps #1-2 SUCCEED
- 2) IF-RECV in step #3 has a ComPacket header value of "All Response(s) returned - no further data", (See [1])
- 3) Steps #4-5 SUCCEED

6.11 Invalid Invoking ID - Get

Notes

Begin Informative Content

The LockingInfo table is a single row table. The UID used in the following test refers to row 5, a nonexistent row of the LockingInfo table.

End Informative Content

Prerequisites

None

Test Sequence

- 1) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
- 2) Invoke Get method on Invoking UID of 00 00 08 01 00 00 00 05
- 3) CLOSE_SESSION

Expected Response

- 1) Steps #1-3 SUCCEED
- 2) The Get method in step #2 and returns an empty results list and a status code of SUCCESS

6.12 Invalid Invoking ID – Non-Get

Notes

Begin Informative Content

The LockingInfo table is a single row table. The UID used in the following test refers to row 5, a non-existing row of the LockingInfo table.

This test uses the Set method to represent all non-Get methods.

End Informative Content

Prerequisites

None

Test Sequence

- 1) Invoke StartSession method with SPID = Locking SP UID
- 2) Invoke Set method on Invoking UID of 00 00 08 01 00 00 00 05
- 3) CLOSE_SESSION

Expected Response

- 1) Steps #1 SUCCEEDS
- 2) The Set method in step #2 Set returns a status code of NOT_AUTHORIZED

- 3) Step #3 SUCCEEDS

6.13 Authorization

Notes

Begin Informative Content

None

End Informative Content

Prerequisites

None

Test Sequence

- 1) Invoke StartSession method with SPID = Locking SP UID
- 2) Invoke Set method on the Enabled column of the User1 Authority
- 3) CLOSE_SESSION

Expected Response

- 1) Steps #1 SUCCEEDS
- 2) The Set method in step #2 Set returns a status code of NOT_AUTHORIZED
- 3) Step #3 SUCCEEDS

6.14 Exceed TPer Properties – Regular Session

Notes

Begin Informative Content

This tests MaxComPacketSize Exceeded. If it is not possible to invoke a Set method that exceeds the TPer's MaxComPacketSize, then this test cannot be performed.

End Informative Content

Prerequisites

None

Test Sequence

- 1) Invoke Properties method to identify the MaxComPacketSize
- 2) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID

- 3) Invoke Set method on the MBR Table such that the ComPacket size exceeds the TPer's MaxComPacketSize
- 4) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
- 5) CLOSE_SESSION

Expected Response

- 1) Steps #1-2
2. IF-RECV in step #3 has a ComPacket header value of "All Response(s) returned - no further data", (See [1])
- 1) Steps #4-5 SUCCEED

6.15 Exceed TPer Properties – Regular Session

Notes

Begin Informative Content

Tests for MaxSubPackets exceeded.

End Informative Content

Prerequisites

None

Test Sequence

- 1) Invoke Properties method to identify the MaxSubPackets
- 2) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
- 3) Send a packet with MaxSubPackets +1 SubPackets. Each SubPacket contains an invocation of the Set method on the DataStore Table
- 4) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
- 5) CLOSE_SESSION

Expected Response

- 1) Steps #1-2 SUCCEED
- 2) IF-RECV in step #3 has a ComPacket header value of "All Response(s) returned - no further data", (See [1])
- 3) Steps #4-5 SUCCEED

6.16 Exceed TPer Properties – Control Session

Notes

Begin Informative Content

Tests for MaxSubPackets exceeded.

End Informative Content

Prerequisites

None

Test Sequence

- 1) Invoke Properties method to identify the MaxSubPackets
- 2) Invoke Properties method with MaxSubPackets +1 SubPackets. Each SubPacket contains an invocation of the Properties Method

Expected Response

- 1) Step #1 SUCCEEDS
- 2) IF-RECV in step #2 has a ComPacket header value of “All Response(s) returned - no further data”, (See [1])

6.17 Overlapping Locking Ranges

Notes

Begin Informative Content

None

End Informative Content

Prerequisites

None

Test Sequence

- 1) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
- 2) Invoke Set method on Locking_Range1. Configure the locking range as follows:
 - a) RangeStart = 0
 - b) RangeLength = 64
 - c) If Opal SSC 2.00 or 2.01, adjust RangeStart and RangeLength according to the RangeAlignment
- 3) Invoke Set method on Locking_Range4. Configure the locking range as follows:

- a) RangeStart = 0
- b) RangeLength = 64
- c) If Opal SSC 2.00 or 2.01, adjust RangeStart and RangeLength according to the RangeAlignment

4) CLOSE_SESSION

Expected Response

- 1) Steps #1-2 SUCCEED
- 2) The Set method in step #3 returns a status code of INVALID_PARAMETER
- 3) Step #4 SUCCEEDS

6.18 Invalid Type

Notes

Begin Informative Content

None

End Informative Content

Prerequisites

None

Test Sequences

- 1) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
- 2) Invoke Set method on the Enabled column of the User1 Authority to value of 0xAAAA
- 3) CLOSE_SESSION

Expected Response

- 1) Steps #1 SUCCEEDS
- 2) The Set method in step #2 returns a status code of INVALID_PARAMETER
- 3) Step #3 SUCCEEDS

6.19 RevertSP – GlobalRange Locked

Notes

Begin Informative Content

None

End Informative Content

Prerequisites

None

Test Sequence

- 1) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
- 2) Invoke Set method on GlobalRange with the following conditions:
 - a) ReadLockedEnabled = TRUE
 - b) WriteLockedEnabled = TRUE
 - c) ReadLocked = TRUE
 - d) WriteLocked = TRUE
- 3) Invoke RevertSP method on the Locking SP with KeepGlobalRangeKey = TRUE
- 4) CLOSE_SESSION

Expected Response

- 1) Steps #1-2 SUCCEED
- 2) Step #3 RevertSP method returns a status code of FAIL
- 3) Step #4 SUCCEEDS

6.20 Activate / ATA Security Interaction

Notes

Begin Informative Content

See [7]

End Informative Content

Prerequisites

- 1) ATA Security Feature Set is enabled
- 2) Locking SP is in the Manufactured-Inactive state

Test Sequence

- 1) Invoke StartSession method with SPID = Admin SP UID and HostSigningAuthority = SID authority UID
- 2) Invoke ACTIVATE method on Locking SP object
- 3) CLOSE_SESSION

Expected Response

- 1) Step #1 SUCCEEDS
- 2) Step #2 ACTIVATE method returns a status code of FAIL
- 3) Step #3 SUCCEEDS

6.21 StartSession on Inactive Locking SP**Notes**

Begin Informative Content

None

End Informative Content

Prerequisites

- 1) Locking SP is in the Manufactured-Inactive state

Test Sequence

- 1) Invoke StartSession method with SPID = Locking SP UID

Expected Response

- 1) The StartSession method in step #1 results in a SyncSession method with a status code of INVALID_PARAMETER

6.22 StartSession with Incorrect HostChallenge**Notes**

Begin Informative Content

None

End Informative Content

Prerequisites

- 1) The C_PIN credential associated with Admin1 has a TryLimit column value of 0; or a Tries column value < the TryLimit column value

Test Sequence

- 1) Invoke StartSession method with SPID = Locking SP UID, HostSigningAuthority = Admin1 authority UID, and HostChallenge = a value that is different from the C_PIN_Admin1 PIN column value

Expected Response

- 1) The StartSession method in step #1 results in a SyncSession method with a status code of INVALID_PARAMETER

6.23 Multiple Sessions**Notes**

Begin Informative Content

None

End Informative Content

Prerequisites

None

Test Sequence

- 1) Invoke StartSession method with SPID = Locking SP UID
- 2) Invoke StartSession method with SPID = Locking SP UID

Expected Response

- 1) Step #1 SUCCEEDS
- 2) The StartSession method in step #2 results in a SyncSession method with a status code of NO_SESSIONS_AVAILABLE or SP_BUSY