SPECIFICATION

TCG Storage Opal Family Test Cases Specification

Specification Version 1.00
Revision 1.00
April 12, 2019

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

TABLES

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; Opalite SSC 1.00; Pyrite SSC 1.00 and 2.00; and Ruby SSC 1.00 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 SSC specifications, 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 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] [10] [11] [12] [13].

1.3 Document References

[1] TCG Storage Architecture Core Specification, Version 2.01

[2] TCG Storage Interface Interactions Specification, Version 1.08

[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)”

[8] TCG Storage Opal SSC Feature Set: Additional DataStore Tables, Version 1.00

[9] TCG Storage Opal SSC Feature Set: PSID, Version 1.00

[10] TCG Storage Security Subsystem Class: Opalite, Version 1.00

[11] TCG Storage Security Subsystem Class: Pyrite, Version 1.00

[12] TCG Storage Security Subsystem Class: Pyrite, Version 2.00

[13] TCG Storage Security Subsystem Class: Ruby, Version 1.00

[14] TCG Storage Feature Set: Block SID Authentication, Version 1.00

1.4 Key Words and Symbols

The key words “MUST,” “MUST NOT,” “REQUIRED,” “SHALL,” “SHALL NOT,” “SHOULD,” “SHOULD NOT,” “RECOMMENDED,” “MAY,” and “OPTIONAL” in this document normative statements are to be interpreted as described in RFC-2119, Key words for use in RFCs to Indicate Requirement Levels.

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.
<LAST_REQUIRED_USER_PASSWORD>	Refers to a 32 byte hex value used as a PIN column value for

	the C_PIN object associated with the last User authority required by each SSC. This Key Word should be interpreted as follows: 1. Opal 1.00: <User4_PASSWORD> 2. Opal 2.00: <User8_PASSWORD> 3. Opal 2.01: <User8_PASSWORD> 4. Opalite 1.00: <User2_PASSWORD> 5. Pyrite 1.00: <User2_PASSWORD> 6. Pyrite 2.00: <User2_PASSWORD> 7. Ruby 1.00: <User2_PASSWORD>
ACE_MBRCONTROL_SET_DONE	Refers to the ACE that grants authorities access to the Set method on the Done column of the MBRControl table. This Key Word should be interpreted as follows: 1. Opal 1.00: ACE_MBRControl_Done 2. Opal 2.00: ACE_MBRControl_DoneToDOR 3. Opal 2.01: ACE_MBRControl_DoneToDOR 4. Opalite 1.00: ACE_MBRControl_DoneToDOR 5. Pyrite 1.00: ACE_MBRControl_DoneToDOR 6. Pyrite 2.00: ACE_MBRControl_DoneToDOR 7. Ruby 1.00: ACE_MBRControl_DoneToDOR
ACTIVATE_THE_LOCKING_SP	Change the life cycle state of the Locking SP in a TPer from Manufactured-Inactive to Manufactured. See [2]
ARBITRARILY_VARYING	Refers to a value that varies between executions in an arbitrary way determined by the Test Suite Vendor.
ARBITRARILY_VARYING_COMMAND_PARAMETERS	Refers to parameters for a command which would normally be considered valid parameters for the command as supported by the SD but that vary between executions in an arbitrary way determined by the Test Suite Vendor.
ARBITRARILY_VARYING_LBA_RANGE	Refers to an LBA range below the Current Maximum LBA that varies between executions in an arbitrary way determined by the Test Suite Vendor.
CLOSE_SESSION	The host transmits an End of Session token.
ENABLE <AuthorityName>	Invoke the Set method to set Enabled column value to TRUE for the noted authority object.
EXPECTED_RESPONSE	See section 2.2.5
FAIL FAILS	Expected failure of one or more test sequence steps.
LAST_REQUIRED_USER	Refers to the last User authority required by each SSC. This Key Word should be interpreted as follows: 1. Opal 1.00: User4 2. Opal 2.00: User8 3. Opal 2.01: User8 4. Opalite 1.00: User2 5. Pyrite 1.00: User2 6. Pyrite 2.00: User2 7. Ruby 1.00: User2
LAST_REQUIRED_RANGE	Refers to the last Locking_* locking object required by each SSC. This Key Word should be interpreted as follows: 1. Opal 1.00: Locking_Range4 2. Opal 2.00: Locking_Range8 3. Opal 2.01: Locking_Range8

	4. Opalite 1.00: Locking_GlobalRange 5. Pyrite 1.00: Locking_GlobalRange 6. Pyrite 2.0: Locking_GlobalRange 7. Ruby 1.00: Locking_GlobalRange
LAST_REQUIRED_RANGE_WRLOCKED_ACE	Refers to the ACE_Locking_*_Set_WrLocked ACE associated with the last Locking_* locking object required by each SSC. This Key Word should be interpreted as follows: 1. Opal 1.00: ACE_Locking_Range4_Set_WrLocked 2. Opal 2.00: ACE_Locking_Range8_Set_WrLocked 3. Opal 2.01: ACE_Locking_Range8_Set_WrLocked 4. Opalite 1.00: ACE_Locking_GlobalRange_Set_WrLocked 5. Pyrite 1.00: ACE_Locking_GlobalRange_Set_WrLocked 6. Pyrite 2.00: ACE_Locking_GlobalRange_Set_WrLocked 7. Ruby 1.00: ACE_Locking_GlobalRange_Set_WrLocked
LAST_REQUIRED_RANGE_RDLOCKED_ACE	Refers to the ACE_Locking_*_Set_RdLocked ACE associated with the last Locking_* locking object required by each SSC. This Key Word should be interpreted as follows: 1. Opal 1.00: ACE_Locking_Range4_Set_RdLocked 2. Opal 2.00: ACE_Locking_Range8_Set_RdLocked 3. Opal 2.01: ACE_Locking_Range8_Set_RdLocked 4. Opalite 1.00: ACE_Locking_GlobalRange_Set_RdLocked 5. Pyrite 1.00: ACE_Locking_GlobalRange_Set_RdLocked 6. Pyrite 2.00: ACE_Locking_GlobalRange_Set_RdLocked 7. Ruby 1.00: ACE_Locking_GlobalRange_Set_RdLocked
MAGIC_PATTERN	A data sequence used in some of the test cases. It has an ARBITRARILY_VARYING value, 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.
NA Not Applicable	Expected behavior or result is not applicable for one or more test sequence steps.
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.
SIZE_OF_MBR_TABLE_DESCRIPTOR_IN_LOGICAL_BLOCKS	The number of logical blocks in the MBR Table. Calculate the number of logical blocks in the MBR Table by dividing the number of rows by the logical block size in bytes obtained through the discovery mechanisms of the underlying interface protocol. The number of rows of the MBR Table can be retrieved by invoking the Get method on the Rows column of the MBR Table Descriptor Object.
SUCCEED	Test Sequence step(s) result in the appropriate response(s) as

SUCCEEDS	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

1.5 Terminology

Table 3 Terminology

TERM	DEFINITION
Current Maximum LBA	The maximum LBA that is permitted at the test run time by a normal read or write command.
Host (or host)	An entity that initiates IF-SEND or IF-RECV to a TPer
IF-RECV	An interface command used by the host to retrieve data from TPer. See [1]
IF-SEND	An interface command used to transmit data from the host to the 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 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 section 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.

Unless otherwise specified, all test cases in this specification apply to all SSCs supported by this specification.

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 SSC Applicability

The SSC Applicability section is a section which will be used to indicate which SSCs are applicable to the test being performed.

If an SSC is identified as not applicable for a given test case, then the Test House SHALL NOT run the test case for the specified SSC and instead the Test House SHALL mark the test as NA.

If this section is omitted from a test case, then the test case applies to all SSCs supported by this specification and the test shall be performed by the Test House for all SSCs, unless otherwise specified.

2.2.3 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.

The prerequisites for each test case SHALL be implemented in sequential order.

2.2.4 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].

2.2.5 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] [8] [9].

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 SPF-07:

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])

d) trigger any TPer resets unless specifically required to do so for a test

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, where 20 is the length of ComPacket header field); 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 SID Original Factory State Requirement

If SID is not MSID, the SD vendor SHALL submit the value of SID to Test House and CPM.

3.3 SSC Version Requirement

The SD vendor SHALL submit the SSC version implemented by the SD to the Test House and CPM.

Supported SSC versions in this specification include:

1) Opal SSC 1.00 (refer to [3])

2) Opal SSC 2.00 (refer to [4])

3) Opal SSC 2.01 (refer to [5])

4) Opalite SSC 1.00 (refer to [10])
5) Pyrite SSC 1.00 (refer to [11])
6) Pyrite SSC 2.00 (refer to [12])
7) Ruby SSC 1.00 (refer to [13])

3.4 Feature Set Support Requirement

The SD vendor SHALL submit a list of Feature Sets implemented by the SD to the Test House and CPM.

The Test Suite Vendor SHALL support testing against all feature sets supported in this specification.

Supported Feature Sets in this specification include:

1) Additional DataStore Tables, Opal SSC Feature Set (refer to [8])
2) PSID, Opal SSC Feature Set (refer to [9])
3) Block SID Authentication Feature Set (refer to [14])

3.5 PSID Feature Set Support Requirement

If the SD vendor claims support for the PSID Feature Set, then the SD vendor SHALL submit the PSID value to the Test House and CPM.

3.6 Interface Read/Write Command Support Requirement

The SD vendor MAY submit a list of all supported Read and Write commands (as identified by [2]) to the Test House and CPM.

The Test Suite SHALL discover the list of all supported Read and Write commands (as identified by [2]).

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

UCT-01: Level 0 Discovery

Notes

Start of informative comment

This test includes the sequence of operations required to determine if the SD supports any SSC supported by this specification.

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

End of informative comment

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. For Opal 1.00, Opal 2.00, Opal 2.01, Opalite 1.00, and Ruby 1.00, Media Encryption = 1
   iii. For Pyrite 1.00 and Pyrite 2.00, Media Encryption = 0
   iv. 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 ComIDs >= 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 ComIDs >= 1
      iii. Number of Locking SP Admin Authorities >= 4
      iv. Number of Locking SP User Authorities >= 8
   b. Geometry Reporting Feature
      i. Feature Code = 0x0003
   c. Additional DataStore Table Feature
      i. Feature Code = 0x0202
      ii. Maximum number of DataStore Tables >= 1
      iii. Maximum total size of DataStore Tables >= 0xA0000
      iv. DataStore Table size alignment >= 1

5) The SD returns the following values for Opalite 1.00:
   a. Opalite SSC 1.00 Feature
      i. Feature Code = 0x0301
      ii. Number of ComIDs >= 1
   b. Block SID Authentication Feature
      i. Feature Code = 0x0402
      ii. SID Value State = 0

6) The SD returns the following values for Pyrite 1.00:
   a. Pyrite SSC 1.00 Feature
      i. Feature Code = 0x0302
      ii. Number of ComIDs >= 1
   b. Block SID Authentication Feature
      i. Feature Code = 0x0402
      ii. SID Value State = 0

7) The SD returns the following values for Pyrite 2.00:

a. Pyrite SSC 2.00 Feature

i. Feature Code = 0x0303

ii. Number of ComIDs >= 1

b. Block SID Authentication Feature

iii. Feature Code = 0x0402

iv. SID Value State = 0

c. Supported Data Removal Mechanism Feature

i. Feature Code = 0x0404

ii. Overwrite Data Erase = 1 or Block Erase = 1

iii. Crypto Erase = 0

8) The SD returns the following values for Ruby SSC 1.00:

a. Ruby SSC 1.00 Feature

i. Feature Code = 0x0304

ii. Number of ComIDs >= 1

b. Block SID Authentication Feature

i. Feature Code = 0x0402

ii. SID Value State = 0

UCT-02: Properties

Notes

Start of informative comment

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

End of informative comment

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 >= 2048 bytes
   b. MaxResponseComPacketSize >= 2048 bytes
   c. MaxPacketSize >= 2028 bytes
   d. MaxIndTokenSize >= 1992 bytes
   e. MaxPackets >= 1
   f. MaxSubpackets >= 1
   g. MaxMethods >= 1
   h. MaxSessions >= 1
   i. MaxAuthentications >= 2
   j. MaxTranactionLimit >= 1
   k. DefSessionTimeout >= 0
3) The SD returns the following values for Host Properties:
   a. MaxComPacketSize >= 2048 bytes and <= 4096 bytes
   b. MaxPacketSize >= 2028 bytes and <= 4076 bytes
   c. MaxIndTokenSize >= 1992 bytes and <= 4040 bytes

UCT-03: Taking Ownership of an SD

Notes

Start of informative comment

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 of informative comment

Prerequisites

None

Test Sequence

1) If Opal SSC 1.00; or if any other SSC supported by this specification 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 any SSC supported by this specification other than Opal 1.00 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 any other SSC supported by this specification and the Initial C_PIN_SID PIN Indicator value = 0, then step #1 SUCCEEDS
2) If any SSC supported by this specification other than Opal 1.00 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

UCT-04: Activate Locking SP when in Manufactured-Inactive State

Notes

Start of informative comment

None

End of informative comment

Prerequisites

1) Locking SP is in the Manufactured-Inactive state
2) The Activate method is implemented

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

UCT-05: Configuring Authorities

Notes

Start of informative comment

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 of informative comment

Prerequisites

None

Test Sequence

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 LAST_REQUIRED_USER
6) SET_PASSWORD_FOR LAST_REQUIRED_USER to <LAST_REQUIRED_USER_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 = LAST_REQUIRED_USER authority UID
13) CLOSE_SESSION

Expected Response

1) Steps #1-13 SUCCEED

UCT-06: Configuring Locking Objects (Locking Ranges)

Notes

Start of informative comment

None

End of informative comment

Prerequisites

None

Test Sequence

1) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
2) This test step varies based on the SSC version:
   a. For Opal, Invoke Set method on LAST_REQUIRED_RANGE. Configure the locking range as follows:
      i. RangeStart = 0
      ii. RangeLength = 64
      iii. ReadLockEnabled = TRUE
      iv. WriteLockEnabled = TRUE
      v. ReadLocked = FALSE
      vi. WriteLocked = FALSE
      vii. For Opal 2.00 and Opal 2.01, LockOnReset **=** {0}
      viii. Adjust RangeStart and RangeLength according to the RangeAlignment
   b. For all SSCs supported by this specification other than Opal**,** Invoke Set method on Locking_GlobalRange. Configure the locking range as follows:
      i. ReadLockEnabled = TRUE
      ii. WriteLockEnabled = TRUE
      iii. ReadLocked = FALSE
      iv. WriteLocked = FALSE
      v. LockOnReset **=** {0}
3) Invoke the Set method on the BooleanExpr column of the LAST_REQUIRED_RANGE_RDLOCKED_ACE ACE object to set the UIDs of the User1 and LAST_REQUIRED_USER Authority objects
4) Invoke the Set method on the BooleanExpr column of the LAST_REQUIRED_RANGE_WRLOCKED_ACE ACE object to set the UIDs of the User1 and LAST_REQUIRED_USER Authority objects
5) CLOSE_SESSION
6) This test step varies based on the SSC version:
   a. For Opal, Write the MAGIC_PATTERN over the entire LAST_REQUIRED_RANGE
   b. For all SSCs supported by this specification other than Opal, Write the MAGIC_PATTERN over an ARBITRARILY_VARYING_LBA_RANGE
7) This test step varies based on the SSC version:
   a. For Opal, Read the entire LAST_REQUIRED_RANGE
   b. For all SSCs supported by this specification other than Opal, Read the same ARBITRARILY_VARYING_LBA_RANGE in Step #6
8) Power cycle the SD
9) This test step varies based on the SSC version:
   a. For Opal, Read the entire LAST_REQUIRED_RANGE

  b. For all SSCs supported by this specification other than Opal, Read the same ARBITRARILY_VARYING_LBA_RANGE in Step #6

10) This test step varies based on the SSC version:

  a. For Opal, Write the MAGIC_PATTERN over the entire LAST_REQUIRED_RANGE

  b. For all SSCs supported by this specification other than Opal, Write the MAGIC_PATTERN over an ARBITRARILY_VARYING_LBA_RANGE

Expected Response

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

UCT-07: Unlocking Ranges

Notes

Start of informative comment

None

End of informative comment

Prerequisites

None

Test Sequence

1) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = User1 authority UID
2) Invoke the Set method on the ReadLocked and WriteLocked columns of the LAST_REQUIRED_RANGE Locking object with a value of FALSE
3) CLOSE_SESSION
4) This test step varies based on the SSC version:
    a. For Opal, Read the entire LAST_REQUIRED_RANGE
    b. For all SSCs supported by this specification other than Opal, Read an ARBITRARILY_VARYING_LBA_RANGE

Expected Results

1) Steps #1-4 SUCCEED

UCT-08: Erasing Ranges

Notes

Start of informative comment

None

End of informative comment

SSC Applicability

This test case applies to all SSCs supported by this specification except for:

1) Pyrite 1.00
2) Pyrite 2.00

Prerequisites

None

Test Sequence

1) This test step varies based on the SSC version:
   a. For Opal, Write the MAGIC_PATTERN over the entire LAST_REQUIRED_RANGE
   b. For all SSCs supported by this specification other than Opal, Write the MAGIC_PATTERN over an ARBITRARILY_VARYING_LBA_RANGE
2) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
3) Invoke the Get method on the LAST_REQUIRED_RANGE to retrieve the ActiveKey column's value
4) Invoke the GenKey method on the UID retrieved from the LAST_REQUIRED_RANGE's ActiveKey column
5) CLOSE_SESSION
6) This test step varies based on the SSC version:
   a. For Opal, Attempt to read the entire LAST_REQUIRED_RANGE
   b. For all SSCs supported by this specification other than Opal, Attempt to read the entire ARBITRARILY_VARYING_LBA_RANGE that was written to in test step #1

Expected Response

1) Steps #1-5 SUCCEED
2) The Read command in step #6 responds in one of the following ways:
   a. The Read command fails without returning data;
   b. The Read command fails and returns data that does not match the MAGIC_PATTERN; or
   c. The Read command succeeds and returns data that does not match the MAGIC_PATTERN

UCT-09: Using the DataStore Table

Notes

Start of informative comment

None

End of informative comment

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

UCT-10: Enable MBR Shadowing

Notes

Start of informative comment

None

End of informative comment

SSC Applicability

This test case applies to all SSCs supported by this specification with the following exception for Pyrite 1.00, Pyrite 2.00, and Ruby 1.00:

This test case only applies to Pyrite 1.00, Pyrite 2.00, and Ruby 1.00 if the MBR Shadowing feature is supported.

Prerequisites

None

Test Sequence

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

2) Invoke the Set method on the BooleanExpr column of the ACE_MBRCONTROL_SET_DONE ACE object to include the UIDs of the User1 and LAST_REQUIRED_USER Authority objects
3) Invoke Get method on the Rows column of the MBR Table Descriptor Object
4) This test step varies based on the SSC version:
   a. For Opal, Invoke the Set method to change the RangeLength column of the LAST_REQUIRED_RANGE to SIZE_OF_MBR_TABLE_DESCRIPTOR_IN_LOGICAL_BLOCKS + 10 LBAs
   b. For Opalite 1.00, Pyrite 1.00, Pyrite 2.00, and Ruby 1.00, Do nothing for this step
5) This test step varies based on the SSC version:
   a. For Opal, Write 1s over the entire LAST_REQUIRED_RANGE
   b. For all SSCs supported by this specification other than Opal, Write 1s over the range from LBA 0 to SIZE_OF_MBR_TABLE_DESCRIPTOR_IN_LOGICAL_BLOCKS + 10
6) This test step varies based on the SSC version:
   a. For Opal 1.00 Invoke Set method to write the entire MBR table with the MAGIC_PATTERN
   b. For all SSCs supported by this specification other than Opal 1.00, Invoke Set method to write the entire MBR table with the MAGIC_PATTERN while adhering to the MandatoryWriteGranularity requirements
7) Invoke Set method on the Enable column of the MBRControl table with a value of TRUE
8) CLOSE_SESSION
9) Power cycle the SD
10) This test step varies based on the SSC version:
    a. For Opal, Write the MAGIC_PATTERN over the entire LAST_REQUIRED_RANGE
    b. For all SSCs supported by this specification other than Opal, Write the MAGIC_PATTERN over the entire range from LBA 0 to SIZE_OF_MBR_TABLE_DESCRIPTOR_IN_LOGICAL_BLOCKS + 10
11) Read from LBA 0 to the size of the MBR Table
12) This test step varies based on the SSC version:
    a. For Opal SSC 1.00 Read 10 LBAs starting immediately following the end of the MBR
    b. For all SSCs supported by this specification other than Opal 1.00, Read 10 LBAs or an appropriate value adhering to the Range Alignment requirements, starting immediately following the end of the MBR Shadow

Expected Response

1) Steps #1-9 SUCCEED
2) Step #10 returns Data Protection Error
3) The value returned from the Read command in step #11 matches the MAGIC_PATTERN
4) The value returned from the Read command in step #12 = 0s

UCT-11: MBR Done

Notes

Start of informative comment

None

End of informative comment

SSC Applicability

This test case applies to all SSCs supported by this specification with the following exception for Pyrite 1.00, Pyrite 2.00, and Ruby 1.00:

This test case only applies to Pyrite 1.00, Pyrite 2.00, and Ruby 1.00 if the MBR Shadowing feature is supported.

Prerequisites

None

Test Sequence

1) Invoke the StartSession method with SPID = Locking SP UID and HostSigningAuthority = LAST_REQUIRED_USER authority UID
2) Invoke the Set method on the ReadLocked and WriteLocked columns of the LAST_REQUIRED_RANGE 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) This test step varies based on SSC version:
   a. For Opal, Read the entire LAST_REQUIRED_RANGE
   b. For all SSCs supported by this specification other than Opal, Read the entire range from LBA 0 to SIZE_OF_MBR_TABLE_DESCRIPTOR_IN_LOGICAL_BLOCKS + 10

Expected Response

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

UCT-12: Revert the Locking SP using SID, with Locking SP in Manufactured state

Notes

Start of informative comment

None

Start of informative comment

Prerequisites

None

Test Sequence

1) Write the MAGIC_PATTERN over 64 logical blocks beginning at LBA 0

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) This test step varies based on the SSC version:
    a. For all SSCs supported by this specification other than Pyrite 1.00, read 64 logical blocks beginning at LBA 0
    b. For Pyrite 1.00, do nothing for this step

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) For all SSCs supported by this specification other than Pyrite 1.00, the Read command in step #6 responds in one of the following ways:
    a. The Read command fails without returning data;
    b. The Read command fails and returns data that does not match the MAGIC_PATTERN; or
    c. The Read command succeeds and returns data that does not match the MAGIC_PATTERN

UCT-13: Revert the Admin SP using SID, with Locking SP in Manufactured-Inactive state

Notes

Start of informative comment

None

End of informative comment

Prerequisites

None

Test Sequence

1) Write the MAGIC_PATTERN over 64 logical blocks beginning at LBA 0
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

    Invoke StartSession method with SPID = Admin SP UID, HostSigningAuthority = SID authority UID, and HostChallenge = C_PIN_MSID PIN column value

    Else

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 64 logical blocks beginning at LBA 0

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

UCT-14: Revert the Admin SP using SID, with Locking SP in Manufactured state

Notes

Start of informative comment

None

End of informative comment

Prerequisites

1) SID's PIN column value is set to <SID_PASSWORD> value in the SID's C_PIN credential PIN column

2) Locking SP is in the Manufactured state

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 64 logical blocks beginning at LBA 0

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

   Invoke StartSession method with SPID = Admin SP UID, HostSigningAuthority = SID authority UID, and HostChallenge = C_PIN_MSID PIN column value

   Else

   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 64 logical blocks beginning at LBA 0

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) For all SSCs supported by this specification other than Pyrite 1.00, The Read command in step #7 responds in one of the following ways:
    a. The Read command fails without returning data;
    b. The Read command fails and returns data that does not match the MAGIC_PATTERN; or
    c. The Read command succeeds and returns data that does not match the MAGIC_PATTERN

UCT-15: Revert Admin SP using Admin1, with Locking SP in Manufactured state

Notes

Start of informative comment

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

End of informative comment

SSC Applicability

This test case applies to Opal 2.00 and Opal 2.01 with no exceptions.

This test case only applies to all other SSCs supported by this specification if the Admin1 authority in the Authority table of the AdminSP is implemented.

Prerequisites

1) Locking SP is in the Manufactured state
2) Admin1 authority is enabled

    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 64 logical blocks beginning at LBA 0
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

    Invoke StartSession method with SPID = Admin SP UID, HostSigningAuthority = SID authority UID, and HostChallenge = C_PIN_MSID PIN column value

    Else

    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 64 logical blocks beginning at LBA 0

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) For all SSCs supported by this specification other than Pyrite 1.00, the Read command in step #7 responds in one of the following ways:
    a. The Read command fails without returning data;
    b. The Read command fails and returns data that does not match the MAGIC_PATTERN; or
    c. The Read command succeeds and returns data that does not match the MAGIC_PATTERN

UCT-16: Revert Admin SP using PSID, with Locking SP in Manufactured state

Notes

Start of informative comment

None

End of informative comment

SSC Applicability

This test case applies to the following SSCs:

1) Opal 2.01
2) Opalite 1.00
3) Pyrite 2.00
4) Ruby 1.00

If the PSID Feature Set is implemented this test case also applies to Opal 1.00 and Opal 2.00.

Prerequisites

1) The PIN column of C_PIN_SID is set to <SID_PASSWORD>
2) Locking SP is in the Manufactured state

Test Sequence

1) Write the MAGIC_PATTERN over 64 logical blocks beginning at LBA 0
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

Invoke StartSession method with SPID = Admin SP UID, HostSigningAuthority = SID authority UID, and HostChallenge = C_PIN_MSID PIN column value

Else

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 64 logical blocks beginning at LBA 0

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 responds in one of the following ways:
    a. The Read command fails without returning data;
    b. The Read command fails and returns data that does not match the MAGIC_PATTERN; or
    c. The Read command succeeds and returns data that does not match the MAGIC_PATTERN

5 Specific Functionality

5.1 Introduction

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

5.2 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

SPF-01: Transaction

Notes

Start of informative comment

There are two tests performed relating to Transactions:

1. Case 1 attempts to write an entire table with the MAGIC_PATTERN.
2. Case 2 attempts to write an entire table with 0s, and then close the session without committing the Transaction.

In most cases, the MBR Table is used for these tests but for SSCs where the MBR shadowing feature is optional, the MBR Table is only used when the MBR shadowing feature is supported, otherwise the DataStore table is used.

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

End of informative comment

Case 1:

Prerequisites

1) For Opal 1.00, Opal 2.00, Opal 2.01, and Opalite 1.00, knowledge of the MBR Table size
2) For Opal 2.00, Opal 2.01, and Opalite 1.00, knowledge of the MandatoryWriteGranularity Column value for the MBR Table
3) For Pyrite 1.00, Pyrite 2.00, and Ruby 1.00, if the MBR Shadowing feature is supported, then knowledge of the MBR Table size, otherwise knowledge of the DataStore Table size
4) For Pyrite 1.00, Pyrite 2.00, and Ruby 1.00, if the MBR Shadowing feature is supported, then knowledge of the MandatoryWriteGranularity Column value for the MBR Table, otherwise knowledge of the MandatoryWriteGranularity Column value for the DataStore Table

Test Sequence

1) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
2) This test step varies based on SSC version:
    a. For Opal 1.00, Invoke Set method to write the entire MBR Table with 0s
    b. For Opal 2.00, Opal 2.01, and Opalite 1.00, Invoke the Set method to write the entire MBR Table with 0s while adhering to the MandatoryWriteGranularity requirements
    c. For Pyrite 1.00, Pyrite 2.00, and Ruby 1.00, if the MBR Shadowing feature is supported, Invoke the Set method to write the entire MBR Table with 0s while adhering to the MandatoryWriteGranularity requirements
    d. For Pyrite 1.00, Pyrite 2.00, and Ruby 1.00, if the MBR Shadowing Feature is not supported, Invoke the Set method to write the entire DataStore Table with 0s while adhering to the MandatoryWriteGranularity requirements
3) CLOSE_SESSION if the write is successful, or if the session aborts due to a timeout, exit the test and record result as NA
4) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
5) Send a subpacket that contains a StartTransaction token with a status code of 0x00
6) This test step varies based on SSC version:
    a. For Opal 1.00, Invoke the Set method to write the entire MBR Table with the MAGIC_PATTERN
    b. For Opal 2.00, Opal 2.01, and Opalite 1.00, Invoke the Set method to write the entire MBR Table with the MAGIC_PATTERN while adhering to the MandatoryWriteGranularity requirements
    c. For Pyrite 1.00, Pyrite 2.00, and Ruby 1.00, if the MBR Shadowing feature is supported, Invoke the Set method to write the entire MBR Table with the MAGIC_PATTERN while adhering to the MandatoryWriteGranularity requirements
    d. For Pyrite 1.00, Pyrite 2.00, and Ruby 1.00, if the MBR Shadowing feature is not supported, Invoke the Set method to write the entire DataStore Table with the MAGIC_PATTERN while adhering to the MandatoryWriteGranularity requirements
7) Send a subpacket that contains an End Transaction token with a status code of 0x00
8) 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
9) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
10) This test step varies based on SSC version:
    a. For Opal 1.00, Opal 2.00, Opal 2.01, and Opalite1.00, Invoke the Get method on the MBR Table to read the data from the table
    b. For Pyrite 1.00, Pyrite 2.00, and Ruby 1.00, if the MBR Shadowing feature is supported, Invoke the Get method on the MBR Table to read the data from the table
    c. For Pyrite 1.00, Pyrite 2.00, and Ruby 1.00, if the MBR Shadowing feature is not supported, Invoke the Get method on the DataStore Table to read the data from the table
11) CLOSE_SESSION

Expected Response

1) Steps #1-11 SUCCEED
2) The Get method in step #10 returns the MAGIC_PATTERN

3) If the session is aborted on step #3 or step #8, the result of this test is NA

Case 2:

Prerequisites

1) Steps #1-11 in Case 1 SUCCEED

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 0x00
3) This test step varies based on SSC version:
   a. For Opal SSC 1.00, Invoke the Set method to write the entire MBR Table with 0s
   b. For Opal SSC 2.00, Opal 2.01, and Opalite 1.00, Invoke the Set method to write the entire MBR Table with 0s while adhering to the MandatoryWriteGranularity requirements
   c. For Pyrite 1.00, Pyrite 2.00, and Ruby 1.00, if the MBR Shadowing feature is supported, Invoke the Set method to write the entire MBR Table with 0s while adhering to the MandatoryWriteGranularity requirements
   d. For Pyrite 1.00, Pyrite 2.00, and Ruby 1.00, if the MBR Shadowing feature is not supported, Invoke the Set method to write the entire DataStore 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) This test step varies based on SSC version:
   a. For Opal 1.00, Opal 2.00, Opal 2.01, and Opalite 1.00, Invoke the Get method on the MBR Table to read the data from the table
   b. For Pyrite 1.00, Pyrite 2.00, and Ruby 1.00, if the MBR Shadowing feature is supported, Invoke the Get method on the MBR Table to read the data from the table
   c. For Pyrite 1.00, Pyrite 2.00, and Ruby 1.00, if the MBR Shadowing feature is not supported, Invoke the Get method on the DataStore Table to read the data from the table.
7) CLOSE_SESSION

Expected Response

1) Steps #1-7 SUCCEED
2) The Get method in step #6 returns the MAGIC_PATTERN
3) If the session is aborted on step #4, the result of this test is NA

SPF-02: IF-RECV Behavior Tests

Notes

Start of informative comment

There are two tests performed relating to IF-RECV Behavior:

Case 1 attempts to issue an IF-RECV command while the SD is in an Awaiting IF-SEND state

Case 2 attempts to issue an IF-RECV command with an Insufficient Transfer Length

End of informative comment

Case 1:

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])

Case 2:

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]

SPF-03: TryLimit

Notes

Start of informative comment

None

End of informative comment

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

   Else do not perform this test step and the Test Suite SHALL mark the result of this step as NA
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

   Else do not perform this test step and the Test Suite SHALL mark the result of this step as NA
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

   Else do not perform this test step and the Test Suite SHALL mark the result of this step as NA

Expected Response

1) Steps #1-7 SUCCEED
2) Steps #8-10 FAIL for any Authority with a TryLimit value >0.
3) Every StartSession method in steps #8a, #9a, and #10a results in a SyncSession method with a status code of NOT_AUTHORIZED
4) StartSession method with the correct HostChallenge value in steps #8b, #9b, and #10b results in a SyncSession method with a status code of AUTHORITY_LOCKED_OUT

SPF-04: Tries Reset

Notes

Start of informative comment

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

End of informative comment

Prerequisites

1) User1 is enabled

Test Sequence

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

e. Invoke Get method on the Tries Column of the User1 Authority's C_PIN Object

f. 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 on steps #8c, #9c, and #10e

SPF-05: Tries Reset on Power Cycle

Notes

Start of informative comment

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

End of informative comment

Prerequisites

1) User1 is enabled

Test Sequence

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

   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

   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

    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 = Admin1 authority UID
  b. Invoke Get method on User1 Authority's C_PIN Tries Column
  c. CLOSE_SESSION

Expected Response

1) Steps #1-7 and steps #11-14 SUCCEED
2) Every StartSession method in steps #8, #9, and #10 results in a SyncSession method with a status code of NOT_AUTHORIZED
3) For test step #12, if SID C_PIN TryLimit Column value > 0, then
  a. Admin SP session opens successfully
  b. Get method on SID Authority's C_PIN Tries Column returns 0
4) For test step #13, if Admin1 C_PIN TryLimit Column value > 0, then
  a. Locking SP session opens successfully
  b. Get method on Admin1 Authority's C_PIN Tries Column returns 0
5) For test step #14, if User1 C_PIN TryLimit Column value > 0, then
  a. Locking SP session opens successfully
  b. Get method on User1 Authority's C_PIN Tries Column returns 0

SPF-06: Next

Notes

Start of informative comment

Testing of the Next method to verify that the method works correctly in multiple conditions.

This test contains two different tests, but only one test is required per SSC, as specified by the SSC applicability section of each test.

End of informative comment

Case 1:

SSC Applicability

This test case applies to the following SSCs:

1) Opal 1.00 SSC
2) Opal 2.00 SSC
3) Opal 2.01 SSC

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 first UID from the list of UIDs returned in step #3, 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 that was second in the list of UIDs returned in Step #3

Case 2:

SSC Applicability

This test case applies to the following SSCs:

1) Opalite 1.00 SSC
2) Pyrite 1.00 SSC
3) Pyrite 2.00 SSC
4) Ruby 1.00 SSC

Prerequisites

None

Test Sequence

1) Invoke StartSession method with SPID = Locking SP UID

2) Invoke Next method on the MethodID Table with an empty parameter list

3) Invoke Next method on the MethodID Table with the Where parameter set to the first UID from the list of UIDs returned in step #3 and the Count parameter set to 1

4) CLOSE_SESSION

Expected Response

1) Steps #1-4 SUCCEED

2) Step #2

   a. returns a list of UIDs where the number of values >= 7, and

   b. the first four bytes of each UID returned are 0x00000006

3) Step #3 returns a list that contains only the second UID from the list of UIDs returned in Step #2

SPF-07: Host Session Number (HSN)

Notes

Start of informative comment

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

End of informative comment

Prerequisites

None

Test Sequence

1) Invoke StartSession method with HostSessionID = ARBITRARILY_VARYING HSN, SPID = Admin SP UID, and HostSigningAuthority = SID 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

SPF-08: RevertSP

Notes

Start of informative comment

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

End of informative comment

Case 1:

SSC Applicability

This test applies to all SSCs supported by this specification.

Prerequisites

None

Test Sequence

1) Write the MAGIC_PATTERN over 64 logical blocks beginning at LBA 0
2) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
3) Invoke RevertSP method with the KeepGlobalRangeKey/KeepData omitted
4) Invoke StartSession method with SPID = Locking SP UID
5) This test step varies based on the SSC version:
   a. For all SSCs supported by this specification other than Pyrite 1.00, read 64 logical blocks beginning at LBA 0
   b. For Pyrite 1.00, do nothing for this step

Expected Response

1) Steps #1-3 SUCCEED
2) The StartSession method in step #4 results in a SyncSession method with a status code of INVALID_PARAMETER
3) For all SSCs supported by this specification other than Pyrite 1.00, the Read command in step #5 responds in one of the following ways:
   a. The Read command fails without returning data;
   b. The Read command fails and returns data that does not match the MAGIC_PATTERN; or
   c. The Read command succeeds and returns data that does not match the MAGIC_PATTERN

Case 2:

SSC Applicability

This test applies to all SSCs supported by this specification except for Pyrite 1.00.

Prerequisites

None

Test Sequence

1) Write the MAGIC_PATTERN over 64 logical blocks beginning at LBA 0
2) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
3) Invoke RevertSP method with the KeepGlobalRangeKey/KeepData present and set to FALSE
4) Invoke StartSession method with SPID = Locking SP UID
5) Read 64 logical blocks beginning at LBA 0

Expected Response

1) Steps #1-3 SUCCEED
2) The StartSession method in step #4 results in a SyncSession method with a status code of INVALID_PARAMETER
3) The Read command in step #5 responds in one of the following ways:
   a. The Read command fails without returning data;
   b. The Read command fails and returns data that does not match the MAGIC_PATTERN; or
   c. The Read command succeeds and returns data that does not match the MAGIC_PATTERN

Case 3:

SSC Applicability

This test applies to all SSCs supported by this specification except for Pyrite 1.00.

Prerequisites

1) Locking_GlobalRange ReadLockEnabled, WriteLockEnabled, ReadLocked and WriteLocked column values = FALSE
2) If non-Global Locking Range objects are implemented, then all non-Global Locking Range objects' ReadLockEnabled, WriteLockEnabled, ReadLocked and WriteLocked column values = FALSE and RangeStart and RangeLength columns = 0

Test Sequence

1) Write the MAGIC_PATTERN over 64 logical blocks beginning at LBA 0
2) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
3) Invoke RevertSP method with the KeepGlobalRangeKey/KeepData present and set to TRUE
4) Invoke StartSession method with SPID = Locking SP UID
5) Read 64 logical blocks beginning at LBA 0

Expected Response

1) Steps #1-3 SUCCEED
2) The StartSession method in step #4 results in a SyncSession method with a status code of INVALID_PARAMETER
3) The Read command in step #5 returns data that matches the MAGIC_PATTERN

SPF-09: Range Alignment Verification

Notes

Start of informative comment

None

End of informative comment

SSC Applicability

This test case only applies to Opal 2.00, Opal 2.01, and Ruby 1.00 if the AlignmentRequired column value in the LockingInfo table = TRUE.

This test case does not apply to any other SSC.

Prerequisites

1) Confirm the AlignmentRequired column value in the LockingInfo table = TRUE. If AlignmentRequired = FALSE do not perform the test and the Test Suite SHALL mark the result as NA.

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 Invoke Set method on RangeLength and RangeStart columns with RangeStart and RangeLength values satisfying the conditions:
    a. [(RangeStart - LowestAlignedLBA) % AlignmentGranularity] = 0
    b. [RangeLength % AlignmentGranularity] = 0
4) CLOSE_SESSION

Expected Response

1) If AlignmentGranularity is = 1 then mark the test NA
2) If AlignmentGranularity is > 1, steps #1-4 SUCCEED

SPF-10: Byte Table Access Granularity

Notes

Start of informative comment

None

End of informative comment

SSC Applicability

This test case only applies to the following SSCs:

1) Opal 2.00
2) Opal 2.01
3) Opalite 1.00
4) Pyrite 1.00
5) Pyrite 2.00
6) Ruby 1.00

Prerequisites

1) Confirm the MandatoryWriteGranularity column value of the DataStore table > 1. If the MandatoryWriteGranularity column value = 1, do not perform the test and the Test Suite SHALL mark the result as NA.

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

SPF-11: Stack Reset

Notes

Start of informative comment

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

End of informative comment

Prerequisites

1) 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 0x00
3) Invoke Set method on the Enabled Column of User1 Authority with a value of TRUE
4) Issue STACK_RESET command
5) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
6) Invoke Get method to retrieve the value of the Enabled Column of User1 Authority
7) CLOSE_SESSION

Expected Response

1) Steps #1-7 SUCCEED
2) The Get method in step #6 returns a value of FALSE

SPF-12: TPer Reset

Notes

Start of informative comment

None

End of informative comment

Case 1:

SSC Applicability

This test case only applies to the following SSCs:

1) Opal 2.00
2) Opal 2.01
3) Opalite 1.00
4) Pyrite 1.00
5) Pyrite 2.00
6) Ruby 1.00

Prerequisites

1) ProgrammaticResetEnable set to TRUE
2) Locking_GlobalRange has ReadLocked and WriteLocked columns set to FALSE
3) Locking_GlobalRange has ReadLockEnabled and WriteLockEnabled columns are set to TRUE
4) LockOnReset column value includes Programmatic
5) For Pyrite 1.00, Pyrite 2.00, and Ruby 1.00, if the MBR Shadowing feature is supported, then the Enable column value of the MBRControl table = FALSE
6) For Opal 1.00, Opal 2.00, 2.01, and Opalite 1.00, the Enable column value of the MBRControl table = FALSE

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 the Locking_GlobalRange ReadLocked and WriteLocked columns
5) CLOSE_SESSION
6) Write the MAGIC_PATTERN over an ARBITRARILY_VARYING_LBA_RANGE
7) Read from the same ARBITRARILY_VARYING_LBA_RANGE as in Step #6

Expected Response

1) Steps #1-5 SUCCEED
2) The Get method in step #4 returns values of TRUE
3) The Write command in Step #5 returns a Data Protection Error
4) The Read command in Step #6 returns a Data Protection Error

Case 2:

SSC Applicability

This test case applies to the following SSCs:

1) Opal 2.00
2) Opal 2.01
3) Opalite 1.00

This test case applies to all other SSCs supported by this specification only if the MBR Shadowing feature is supported.

Prerequisites

1) ProgrammaticResetEnable column is set to TRUE
2) Locking_GlobalRange has ReadLockEnabled and WriteLockEnabled columns are set to FALSE
3) DoneOnReset column value includes Programmatic
4) Done column is set to TRUE
5) First 64 logical blocks of MBR Table has been set with MAGIC_PATTERN
6) If the MBR Shadowing feature is supported, the Enable column value of the MBRControl table = TRUE

Test Sequence

1) Write 0s to 64 logical blocks beginning at LBA 0
2) Issue the TPER_RESET command
3) Write 1s to 64 logical blocks beginning at LBA 0
4) Read 64 logical blocks beginning at LBA 0

5) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID.
6) Invoke the Get method on the MBRControl Done column

Expected Response

1) Steps #1-2 SUCCEED
2) The Write command in Step #3 returns a Data Protection Error
3) The Read command in Step #4 returns data that matches the MAGIC_PATTERN
4) Steps #5-6 SUCCEED
5) The Get method in step #6 returns the value of FALSE

SPF-13: Authenticate

Notes

Start of informative comment

None

End of informative comment

SSC Applicability

This test case only applies to the following SSCs:

1) Opal 2.00
2) Opal 2.01
3) Opalite 1.00
4) Pyrite 1.00
5) Pyrite 2.00
6) Ruby 1.00

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

SPF-14: Session Abort (Deprecated)

This test case has been removed due to similar functionality being tested elsewhere. This section MAY be removed in a future version of this specification.

SPF-15: Random

Notes

Start of informative comment

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

End of informative comment

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

SPF-16: CommonName

Notes

Start of informative comment

None

End of informative comment

SSC Applicability

This test case only applies to the following SSCs:

1) Opal 2.00
2) Opal 2.01
3) Ruby 1.00

Prerequisites

1) Admin1 is enabled

Test Sequence

1) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
2) Invoke the Set method on the CommonName column of the Admin1 authority object using the MAGIC_PATTERN
3) Invoke the Set method on the CommonName column of Locking_GlobalRange using the MAGIC_PATTERN
4) Invoke Get method on the CommonName column of the Admin1 authority object
5) Invoke Get method on the CommonName column of Locking_GlobalRange
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

SPF-17: Additional DataStore Tables

Notes

Start of informative comment

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 of informative comment

Case 1:

SSC Applicability

This test case applies to the following SSCs:

1) Opal 2.00 and 2.01
2) All other SSCs supported by this specification, if the Additional DataStore Tables Feature Set is implemented

Prerequisites

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

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 = 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

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

Case 2:

SSC Applicability

This test case applies to the following SSCs:

1) Opal 2.00 and 2.01
2) All other SSCs supported by this specification, if the Additional DataStore Tables Feature Set is implemented

Prerequisites

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

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 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 each DataStore table's Rows column value from the Table table
7) CLOSE_SESSION

Expected Response

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

SPF-18: Range Crossing Behavior

Notes

Start of informative comment

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 of informative comment

SSC Applicability

This test case applies to the following SSCs:

1) Opal 1.00, 2.00 and 2.01
2) All other SSCs supported by this specification, if Locking_Range1 is implemented

Prerequisites

1) Locking_Range1 length is non-zero and does not span the entire SD
2) Locking_GlobalRange and Locking_Range1 are unlocked

Test Sequence

1) Issue a Write command with the MAGIC_PATTERN, with a beginning LBA in Locking_Range1 and ending LBA in Locking_GlobalRange
2) Issue a Read command, with a beginning LBA in Locking_Range1 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

SPF-19: Block SID Authentication

Notes

Start of informative comment

None

End of informative comment

SSC Applicability

This test case applies to the following SSCs:

1) Opalite 1.00
2) Pyrite 1.00
3) Pyrite 2.00

4) Ruby 1.00
5) All other SSCs supported by this specification, if the Block SID Feature Set is implemented

Prerequisites

1) SID C_PIN credential is the same as the value of the MSID C_PIN credential

Test Sequence

1) Issue IF-SEND with the following parameters:
   a. Protocol ID = 0x02
   b. ComID = 0x0005
   c. Hardware Reset bit in Clear Events field = 1
2) Invoke StartSession method with SPID = Admin SP UID and HostSigningAuthority = SID authority UID.
3) 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
4) Trigger a TCG Storage Hardware Reset on the SD
5) Invoke StartSession method with SPID = Admin SP UID and HostSigningAuthority = SID authority UID.
6) Issue IF-SEND with the following parameters:
   a. Protocol ID = 0x02
   b. ComID = 0x0005
   c. Hardware Reset = 0
7) Invoke StartSession method with SPID = Admin SP UID and HostSigningAuthority = SID authority UID.
8) 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
9) Power cycle the SD
10) Invoke StartSession method with SPID = Admin SP UID and HostSigningAuthority = SID authority UID.

Expected Response

1) Steps #1 and #3-10 SUCCEED
2) The StartSession method in step #2 results in a SyncSession method with a status code of NOT_AUTHORIZED
3) The Hardware Reset field in the Block SID Authentication Feature Descriptor in the Level 0 Discovery response returned in #3 = 1
4) The StartSession method in step #7 results in a SyncSession method with a status code of NOT_AUTHORIZED

5) The Hardware Reset field in the Block SID Authentication Feature Descriptor in the Level 0 Discovery response returned in #8 = 0

SPF-20: Data Removal Mechanism

Start of informative comment

Test Set and Get on the ActiveDataRemovalMechanism column in the Data Removal Mechanism table to make sure this table is functional

End of informative comment

SSC Applicability

This test case applies to the following SSCs:

1) Pyrite 2.00

Prerequisites

1) Knowledge of supported Data Removal Mechanisms from Supported Data Removal Mechanisms Feature Descriptor in Level 0 Discovery

Test Sequence

1) Invoke the StartSession method with SPID = Admin SP UID and HostSigningAuthority = SID authority UID.
2) Invoke the Get method on the ActiveDataRemovalMechanism column of the DataRemovalMechanism table
3) Invoke the Set method on the ActiveDataRemovalMechanism column of the DataRemovalMechanism table with one of the Support Data Removal Mechanisms returned in Level 0 Discovery
4) CLOSE_SESSION
5) Invoke the StartSession method with SPID = Admin SP UID and HostSigningAuthority = Anybody authority UID.
6) Invoke the Get method on the ActiveDataRemovalMechanism column of the DataRemovalMechanism table
7) CLOSE_SESSION

Expected Response

1) Steps #1-7 SUCCEED
2) The value returned from the Get method in Step #2 matches is equal to one of the bit set in the Supported Data Removal Mechanisms returned in Level 0 Discovery
3) The value returned from the Get method in Step #6 matches the value that was set in Step #3

6 Error Test Cases

6.1 Introduction

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 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.

6.2 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 sessions are Read-Write sessions
5. No sessions are open

ETC-01: Native Protocol Read/Write Locked Error Responses

Notes

Start of informative comment

None

End of informative comment

Prerequisites

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

2) If non-Global Locking Range objects are implemented, then all non-Global Locking Range objects ReadLockEnabled, WriteLockEnabled, ReadLocked and WriteLocked column values = FALSE and RangeStart and RangeLength columns values = 0

3) For Pyrite 1.00, Pyrite 2.00, and Ruby 1.00, if the MBR Shadowing feature is supported, then the Enable column value of the MBRControl table = FALSE

4) For Opal 1.00, Opal 2.00, 2.01, and Opalite 1.00, the Enable column value of the MBRControl table = FALSE

Test Sequence

1) Issue each of the Write commands (as identified by [2]) that are supported by the SD and the Test Suite. If an LBA range is required for a supported command, write to an ARBITRARILY_VARYING_LBA_RANGE. If other

parameters are required for a supported command, use ARBITRARILY_VARYING_COMMAND_PARAMETERS. Refer to section 3.6

2) Issue each of the Read commands (as identified by [2]) that are supported by the SD and the Test Suite. If an LBA range is required for a supported command, read from an ARBITRARILY_VARYING_LBA_RANGE. If other parameters are required for a supported command, use ARBITRARILY_VARYING_COMMAND_PARAMETERS. Refer to section 3.6

Expected Response

1) Each of the issued commands in Steps #1-2 FAIL
2) For all supported Write commands in step #1 and all supported Read commands in step #2, the SD SHALL:
   a. Transfer no data
   b. Return a Data Protection Error, (See [2])

ETC-02: General – IF-SEND/IF-RECV Synchronous Protocol

Notes

Start of informative comment

None

End of informative comment

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

ETC-03: Invalid IF-SEND Transfer Length

Notes

Start of informative comment

None

End of informative comment

Prerequisites

None

Test Sequence

1) Invoke Properties method to determine SD's MaxComPacketSize
2) Invoke Properties method with the correct ComPacket Header Length field to match the required ComPacket payload size but with the IF-SEND Transfer Length set to a value > MaxComPacketSize

Expected Responses

1) Step #1 SUCCEEDS
2) The IF-SEND in step #2 fails with a result of "Invalid Transfer Length parameter on IF-SEND"

ETC-04: Invalid SessionID - Regular Session

Notes

Start of informative comment

None

End of informative comment

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])

ETC-05: Unexpected Token Outside of Method – Regular Session

Notes

Start of informative comment

This test verifies the condition corresponding to [1], Section 3.2.2.4.2 item 2. The reason on the expected response #2 for "All Response(s) returned - no further data" is because of the device is in the "Awaiting IF_SEND" state, see [1], Section 3.3.10.5

End of informative comment

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

ETC-06: Unexpected Token in Method Header – Regular Session

Notes

Start of informative comment

This test verifies the condition corresponding to [1], Section 3.2.2.4.2 item 3.

End of informative comment

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, or returns a ComPacket with a CloseSession method.
3) Step #3 SUCCEEDS if step #2 returns NOT_AUTHORIZED

ETC-07: Unexpected Token Outside of Method – Control Session

Notes

Start of informative comment

None

End of informative comment

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

ETC-08: Unexpected Token in the Method Parameter List – Control Session

Notes

Start of informative comment

This test verifies the condition corresponding to [1], Section 3.2.2.4.2 items 1 and 4. The reason on the expected response #1a, for “All Response(s) returned - no further data” is because of the description in [1], Section 3.3.7.1.5: “The Host or TPer is free at any time to end a session in which it is participating, but only the host SHALL end the session successfully.”

End of informative comment

Prerequisites

None

Test Sequence

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

Expected Response

1) One of the following responses is generated:
    a. IF-RECV in step #1 has a ComPacket header value of “All Response(s) returned - no further data”, (See [1])
    b. The Properties method in step #1 returns INVALID_PARAMETER

ETC-09: Exceeding Transaction Limit

Notes

Start of informative comment

None

End of informative comment

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]), or returns a ComPacket with a CloseSession method.
3) Steps #4-5 SUCCEED

ETC-10: Invalid Invoking ID - Get

Notes

Start of informative comment

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.

This test case tests the following requirement from [1]:

Unless otherwise noted in a method's description, this status code (NOT_AUTHORIZED) SHALL be returned whenever there is no row in the AccessControl table to represent the InvokingID/MethodID combination, or when there is a row but the ACL for the InvokingID/MethodID combination has not been satisfied.

End of informative comment

Case 1:

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 AA BB CC DD
3) CLOSE_SESSION

Expected Response

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

Case 2:

Notes

Start of informative comment

This test validates correct behavior when the Get method is invoked on a Byte Table and the authority does not have access to retrieve contents from the byte table.

This test case tests the following requirement from [1]:

If the currently authenticated authorities do not satisfy the access control restrictions for invoking Get on a byte table, the method SHALL return an empty results list.

End of informative comment

Prerequisites

None

Test Sequence

1) Invoke StartSession method with SPID = Locking SP UID and HostSigningAuthority = Anybody authority UID
2) Invoke Get method on Invoking UID of 00 00 10 01 00 00 00 00 (DataStore Table)
3) CLOSE_SESSION

Expected Response

1) Steps #1 SUCCEEDS
2) The Get method in step #2 returns a status code of NOT_AUTHORIZED or SUCCESS and an empty results list
3) Step #3 SUCCEEDS

Case 3:

Notes

Start of informative comment

This test validates correct behavior when the Get method is invoked on an Object Table and the authority does not have access to retrieve contents from the Object table.

This test case tests the following requirement from [1]:

When the Get method is invoked on a table or object, only the values that are readable based on currently authenticated authorities and their associated ACE restrictions for the method SHALL be returned.

Cell values that have been requested but are not permitted to be read by the currently authenticated authorities are not returned. Since the return value of the method for non-byte tables is a list of namevalue pairs, cells to which the host invoking the Get method does not have access are omitted from the return result. If a column is known to exist but not returned with a value, then the host is able to discern that it did not have permission to invoke Get on that cell. It is not an error to request columns that are not permitted to be retrieved.

End of informative comment

Prerequisites

None

Test Sequence

1) Invoke the StartSession method with SPID = Locking SP UID and HostSigningAuthority = Admin1 authority UID
2) Invoke the Get method on the InvokingID 00 00 00 0B 00 01 00 01 (C_PIN_Admin1) to get the PIN, CharSet, TryLimit, and Tries columns.
3) CLOSE_SESSION

Expected Response

1) Steps #1 SUCCEEDS
2) The Get method in step #2 returns a status code of SUCCESS and only returns the CharSet, TryLimit, and Tries column values.
3) Step #3 SUCCEEDS

Case 4:

Notes

Start of informative comment

This test validates correct behavior when the Get method is invoked on a non-Table UID.

This test case is similar to Test Case 1, but instead this test case tests with a valid InvokingUID but there is no row in the ACL table that matches the InvokingID/MethodID combination.

This test case tests the following requirement from [1]:

Unless otherwise noted in a method's description, this status code (NOT_AUTHORIZED) SHALL be returned whenever there is no row in the AccessControl table to represent the InvokingID/MethodID combination, or when there is a row but the ACL for the InvokingID/MethodID combination has not been satisfied.

End of informative comment

Prerequisites

None

Test Sequence

1) Invoke the StartSession method with SPID = Locking SP UID and HostSigningAuthority = Anybody authority UID
2) Invoke the Get method on the InvokingID 00 00 00 00 00 00 00 01 (ThisSP)

3) CLOSE_SESSION

Expected Response

1) Steps #1 SUCCEEDS
2) The Get method in step #2 returns a status code of NOT_AUTHORIZED and an empty results list
3) Step #3 SUCCEEDS

ETC-11: Invalid Invoking ID – Non-Get

Notes

Start of informative comment

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 of informative comment

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 returns a status code of NOT_AUTHORIZED
3) Step #3 SUCCEEDS

ETC-12: Authorization

Notes

Start of informative comment

None

End of informative comment

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

ETC-13: Malformed ComPacket Header – Regular Session

Notes

Start of informative comment

This tests a malformed Length field in the ComPacket header whereas TRANSFER LENGTH field in IF-SEND CDB has a correct value. If it is not possible to invoke a Set method that exceeds the TPer's MaxComPacketSize, then this test cannot be performed and the result should be marked as NA.

End of informative comment

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) This test step varies based on SSC version:
   a. For Opal 1.00, invoke Set method on the MBR Table, such that the Length field in the ComPacket header exceeds the TPer's MaxComPacketSize – 20 (where 20 is the length of ComPacket header field), and the IF-SEND Transfer Length set to a value <= MaxComPacketSize
   b. For all SSCs supported by this specification other than Opal 1.00, invoke Set method on the Datastore Table such that the Length field in the ComPacket header exceeds the TPer's MaxComPacketSize – 20 (where 20 is the length of ComPacket header field), and the IF-SEND Transfer Length set to a value <= MaxComPacketSize
4) Issue IF-RECV

Expected Response

1) Steps #1-2 SUCCEED
2) The IF_SEND in step #3:
   a. SUCCEEDS; or
   b. FAILS with a result of "Invalid Transfer Length parameter on IF-SEND"

3) The IF-RECV in step #4 returns a ComPacket header with a value of “All Response(s) returned - no further data” (See [1]), or returns a ComPacket with a CloseSession method.

ETC-14: Exceed TPer Properties – Regular Session

Notes

Start of informative comment

Tests for MaxSubPackets exceeded.

This test verifies the condition corresponding to [1], Section 5.2.2.4.1.1. The reason on the expected response #1a, for “All Response(s) returned - no further data” is because of the description in [1], Section 3.3.7.1.5: “The Host or TPer is free at any time to end a session in which it is participating, but only the host SHALL end the session successfully.”

End of informative comment

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]), or returns a ComPacket with a CloseSession method.
3) Steps #4-5 SUCCEED

ETC-15: Exceed TPer Properties – Control Session

Notes

Start of informative comment

Tests for MaxSubPackets exceeded.

End of informative comment

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])

ETC-16: Overlapping Locking Ranges

Notes

Start of informative comment

None

End of informative comment

SSC Applicability

This test case applies to the following SSCs:

1) Opal 1.00, 2.00 and 2.01
2) All other SSCs supported by this specification, if the MaxRanges column value in the LockingInfo table is > 1

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 any SSC supported by this specification other than Opal 1.00, adjust RangeStart and RangeLength according to the RangeAlignment
3) Invoke Set method on Locking_Range2. Configure the locking range as follows:
    a. RangeStart = 0
    b. RangeLength = 64
    c. If any SSC supported by this specification other than Opal 1.00, 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

ETC-17: Invalid Type

Notes

Start of informative comment

None

End of informative comment

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

ETC-18: RevertSP – GlobalRange Locked

Notes

Start of informative comment

None

End of informative comment

SSC Applicability

This test case applies to all SSCs supported by this specification other than Pyrite 1.00

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/KeepData = 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

ETC-19: Activate / ATA Security Interaction

Notes

Start of informative comment

See [7]

End of informative comment

Prerequisites

1) SID's PIN column value is set to <SID_PASSWORD> value in the SID's C_PIN credential PIN column
2) The ATA Security Feature Set Security Characteristics are Enabled and Unlocked
3) 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

ETC-20: StartSession on Inactive Locking SP

Notes

Start of informative comment

None

End of informative comment

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

ETC-21: StartSession with Incorrect HostChallenge

Notes

Start of informative comment

None

End of informative comment

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 NOT_AUTHORIZED

ETC-22: Multiple Sessions

Notes

Start of informative comment

There are two tests performed regarding multiple sessions:

Case 1 attempts to start two Read-Write sessions with the Locking SP

Case 2 attempts to start MaxSessions + 1 Read-Only sessions with the Locking SP

End of informative comment

Case 1:

Prerequisites

1) No open sessions exist at the start of the Test Sequence

Test Sequence

1) Invoke Properties method to identify the MaxSessions
2) Invoke StartSession method with SPID = Locking SP UID and Write = TRUE
3) Invoke StartSession method with SPID = Locking SP UID and Write = TRUE

Expected Response

1) Step #1-2 SUCCEEDS
2) The StartSession method in step #3 results in a SyncSession method with a status code of:
    a) If MaxSessions = 1: SP_BUSY or NO_SESSIONS_AVAILABLE
    b) If MaxSession <> 1: SP_BUSY

Case 2:

Prerequisites

1) No open sessions exist at the start of the Test Sequence

Test Sequence

1) Invoke Properties method to identify the MaxSessions and MaxReadSessions. If MaxSessions = 0 or MaxReadSessions = 0 or MaxReadSessions is omitted, do not perform this test and the Test Suite SHALL mark the result as NA
2) Invoke the StartSession method with SPID = Locking SP UID and Write = FALSE up to the lesser of MaxSessions or MaxReadSessions
3) Invoke StartSession method with SPID = Locking SP UID and Write = FALSE

Expected Response

1) Step #1 SUCCEEDS

2) Every StartSession method invoked in step #2 results in a SyncSession method with a status code of SUCCESS
3) The StartSession method in step #3 results in a SyncSession method with a status code of NO_SESSIONS_AVAILABLE

ETC-23: Data Removal Mechanism – Set Unsupported Value

Notes

Start of informative comment

Test Set on the ActiveDataRemovalMechanism column in the Data Removal Mechanism table with an invalid value to make sure a proper error is returned

End of informative comment

SSC Applicability

This test case applies to the following SSCs:

1) Pyrite 2.00

Prerequisites

1) Knowledge of supported Data Removal Mechanisms from Supported Data Removal Mechanisms Feature Descriptor in Level 0 Discovery

Test Sequence

1) Invoke the StartSession method with SPID = Admin SP UID and HostSigningAuthority = SID authority UID
2) Invoke the Set method on the ActiveDataRemovalMechanism column of the DataRemovalMechanism table with a value that is not one of the Support Data Removal Mechanisms returned in Level 0 Discovery
3) CLOSE_SESSION

Expected Response

1) Steps #1-3 SUCCEED
2) The Set method in Step #2 returns INVALID_PARAMETER