TCG Storage
Opal Test Cases Specification

Version 1.00 FINAL
Revision 1.00

04 February, 2011

Contact: admin@trustedcomputinggroup.org

TCG Published

Change History

Version 1.00	Date	Description
Rev 1.00	04 February, 2011	First publication

1. Introduction

1.1. Scope

The test cases do not define tests where multiple errors are encoded in a payload from the host. If multiple errors are encoded, the device SHALL take the appropriate action based on one of the errors that occurred.

1.2. References

[1] TCG Storage Architecture Core Specification, Version 2.00
[2] TCG Storage Workgroup Interface Interactions Specification, Version 1.0
[3] TCG Storage Security Subsystem Class: Opal, Version 1.00
[4] Internet Engineering Task Force (IETF), "Key words for use in RFCs to Indicate Requirement Levels" (RFC 2119)
[5] [INCITS T13/2015-D], “Information technology - ATA/ATAPI Command Set – 2 (ACS-2)”

2. Global Notes

2.1. Terminologies

Key words are used to signify the requirements in the specification. The key words “SHALL,” "SHALL NOT," “MAY,” and “OPTIONAL” are used in this document in the same manner as [1]. These key words are to be interpreted as described in [4].

Regarding ATA devices, the difference between the use of TRUSTED SEND or TRUSTED SEND DMA SHALL NOT produce a different results in test. This also applies to the use of TRUSTED RECEIVE and TRUSTED RECEIVE DMA.
It is intended both PIO and DMA commands to be tested for each test.

In this document, the phrase "Security Protocol" and "Protocol ID" refer to the same thing.
Likewise, the phrase "Security Protocol Specific" and "ComID" refer to the same thing.
"Security Protocol" and "Security Protocol Specific" are used in the descriptions that refer to SCSI or ATA interface level behavior.
"Protocol ID" and "ComID" are used in the description that refer to behaviors defined in TCG.

2.2. TPer Life Cycle State

Unless otherwise specifically stated in the description, the following preconditions as to the lifecycle state of the Locking SP are intended.

- If Locking SP's initial lifecycle state is Manufactured-Inactive, the test SHALL be run with the target device in its OFS with the exception of Activating the Locking SP.
- If Locking SP's initial lifecycle state is Manufactured, the test SHALL be run with the target device in its OFS.

2.3. TPer Capabilities

Unless specifically stated in the description, the tests in this document are intended to be performed when the TPer’s communication properties are such that:

- Synchronous Interface Communications capability (see section 3.3.10 in the CoreSpec[1]) is being used for host to TPer communications;
- Buffer Management capability (see section 3.3.8.2 in the CoreSpec[1]) is not being used; and,
- Ack/Nak capability (see section 3.3.9 in the CoreSpec[1]) is not being used.

2.4. IF-SEND/IF-RECV Use

Unless otherwise specifically stated in the description, the tests in this document are intended to be performed with conditions as follows:

- if it is necessary to send/receive token stream,
  - Security Protocol is 1;
  - Security Protocol Specific is any static ComID the TPer supports and as reported by the Opal SSC Feature Descriptor;
  - Transfer Length is the minimum value necessary to transfer a ComPacket; and,
  - when a host sends token stream to the TPer:
    - such ComPacket that:
      - its header's:
        - 'Reserved' field is set to all-0s;
        - 'ComID' field is set to the same value as the Security Protocol field in the IF-SEND;
        - 'ComID Extension' field is set to all-0s;
        - 'OutstandingData' field is set to all-0s;
        - 'MinTransfer' field is set to all-0s; and,
        - 'Length' field is set to a multiple-of-4 value that does not exceed (the TPer's MaxComPacketSize – 20), and,
      - its payload contains exactly one Packet;
    - such Packet that:
      - its header's:
        - 'Session' field is set to:
          - all-0s for Control session, or,
          - the session number of the session that was successfully started by a StartSession() and a SyncSession() for Regular session;
        - 'SeqNumber' is set to all-0s;
        - 'Reserved' field is set to all-0s;
        - 'AckType' is set to all-0s;
        - 'Acknowledgement' field is set to all-0s; and,
        - 'Length' field is set to a multiple-of-4 value that does not exceed (the TPer's MaxPacketSize – 24); and,
      - its payload contains exactly one Subpacket and one Pad field, if necessary;
    - such Subpacket that:
      - its header's:
        - 'Reserved' field is set to all-0s;
        - 'Kind' field is set to all-0s; and,
        - 'Length' field is set to such value that is exactly the length of token stream the host is sending to the TPer; and,
    - such Pad field that:

- its length is zero to three; and,
- its value is set to all-0s.

➢ when a host receives a token stream, including the case where the host receives ComPacket/Packet/Subpacket header(s) only from the TPer, the conditions described in 2.5.1 SHALL be satisfied in each test:

- if a test requires examining the Interface Command Parameters or ComPacket/Packet/Subpacket headers with values other than described above, the test SHALL contain a payload that would not cause errors or state changes within the TPer (e.g. invocation of the Properties method).
- if Regular sessions are necessary, Read-Write sessions:
  - ➢ with:
    - ✧ SID authority authentication using MSID credential for sessions to the Admin SP, or,
    - ✧ Admin1 authority authentication using MSID credential for sessions to the Locking SP; and,
- If it is necessary to do a Stack Reset:
  - ➢ Security Protocol is 2;
  - ➢ Security Protocol Specific is any static ComID the TPer supports and as reported by the Opal SSC Feature Descriptor;
  - ➢ Transfer Length is the minimum value necessary to transfer a StackReset request/response; and,
  - ➢ when a host sends an IF-SEND to the TPer, ‘ComID’ field in the payload is set to the same value as the Security Protocol field in the IF-SEND.

2.5. IF-RECV payload Checking

2.5.1. Security Protocol = 1, Security Protocol Specific = a supported static ComID

The following conditions SHALL be satisfied in each test.

- If the test expects the TPer to be the Awaiting IF-SEND state for that static ComID, a), see 2.5.1.1, SHALL be satisfied.
- If the test expects the TPer to be the Awaiting IF-RECV state for that static ComID, either b), c), d) or e), see 2.5.1.2 to 2.5.1.5, SHALL be satisfied, but:
  - ➢ when the payload satisfies b), the host SHALL re-issue an IF-RECV until the TPer returns a payload that does not satisfy b),
  - ➢ when the payload satisfies c), the host SHALL issue another IF-RECV with greater value of Transfer Length than the previous until the TPer returns a payload that does not satisfy c), and,
  - ➢ when the payload satisfies d), the host SHALL issue an IF-RECV until the TPer returns a payload that does not satisfy d).

2.5.1.1. a) Awaiting IF-SEND state

- ✧ ComPacket header:
  - ➢ ‘Reserved’ field SHALL be set to all-0s;
  - ➢ ‘ComID’ field SHALL be set to the same value as the Security Protocol Specific field in the IF-RECV;
  - ➢ ‘ComID Extension’ field SHALL be set to all-0s;
  - ➢ ‘OutstandingData’ field SHALL be set to all-0s;
  - ➢ ‘MinTransfer’ field SHALL be set to all-0s; and,
  - ➢ ‘Length’ field SHALL be set to all-0s,

2.5.1.2. b) Processing state

The same conditions as a), see 2.5.1.1, with an alternative noted as follows:

- ComPacket header:
  - ‘OutstandingData’ field SHALL be set to 1.

2.5.1.3. c) "Response ready, insufficient transfer length requested"

The same conditions as a), see 2.5.1.1, with alternatives noted as follows:

- ComPacket header:
  - ‘OutstandingData’ field SHALL be set to <total data available>; and,
  - ‘MinTransfer’ field SHALL be set to <minimum request required to transfer a packet>.

2.5.1.4. d) "Response, additional Response(s) available"

- ComPacket header:
  - ‘Reserved’ field SHALL be set to all-0s;
  - ‘ComID’ field SHALL be set to the same value as the Security Protocol field in the IF-RECV;
  - ‘ComID Extension’ field SHALL be set to all-0s;
  - ‘OutstandingData’ field SHALL be set to <additional bytes available, not including the data transferred in the current ComPacket>;
  - ‘MinTransfer’ field SHALL be set to <minimum request required to transfer the next packet>; and,
  - ‘Length’ field SHALL be set to a value which satisfies the following conditions.
    - greater than or equal to the size of Packet header, 24, plus the size of Subpacket, 12,
    - multiple of 4,
    - does not exceed (the Host’s MaxComPacketSize – the size of ComPacket header, 20), and,
    - does not exceed the length of IF-RECV payload
- Packet header:
  - ‘Session’ field SHALL be set to the value the method was invoked or transaction start/end request or end session request was sent with an alternative noted as follows:
    - when a CloseSession is sent, this field SHALL be set to all-0s;
  - ‘SeqNumber’ field SHALL be set to all-0s;
  - ‘Reserved’ field SHALL be set to all-0s;
  - ‘AckType’ field SHALL be set to all-0s;
  - ‘Acknowledgement’ field SHALL be set to all-0s; and,
  - ‘Length’ field SHALL be set to a value which satisfies the following conditions.
    - greater than the value of Length field in the ComPacket header – the size of Packet header, 24, and,
    - does not exceed (the Host’s MaxPacketSize – the size of Packet header, 24),
- Subpacket header:
  - ‘Reserved’ field SHALL be set to all-0s;

- ‘Kind’ field SHALL be set to all-0s; and,
- ‘Length’ field is set to (the value in the Length field in the Packet header – the size of Subpacket header,12) or minus 1,2 or 3 of that,

✧ Subpacket payload:
- See 2.5.1.6.

2.5.1.5. e) "All Response(s) returned, no further data"

The same conditions as d), see 2.5.1.4, with alternatives noted as follows:

✧ ComPacket header:
- ‘OutstandingData’ field SHALL be set to all-0s; and,
- ‘MinTransfer’ field SHALL be set to all-0s.

2.5.1.6. Token Stream

Empty atoms in the returned token stream SHALL be discarded when the following conditions are examined.
The length of each token, including its header field, SHALL NOT exceed the Host’s MaxIndTokenSize.

If the host sends a method invocation or Transaction Start/End request or End of Session request in a way that requires the TPer to discard and/or ignore the request, no tokens SHALL be returned for that invocation or request.
If a method invocation or Transaction Start/End request is expected to cause session abort, no tokens MAY be returned for that invocation or request. If the TPer supports CloseSession(), it MAY be returned additionally. In that case the CloseSession() SHALL satisfy the conditions in Table 6.
If a method invocation or Transaction Start/End request or End Of Session request is expected to be responded with some token(s),the response(s) SHALL be in the order of the invocation or/and request(s) happened, and the following conditions SHALL be satisfied.

- Every method response or transaction control response retrieved from the TPer SHALL be completely contained within a Subpacket, and not split across Subpackets.
- For Properties() invocations with HostProperties, with the expectation that the1st status in the response is SUCCESS, see Table 1.
- For Properties() invocations without HostProperties, with the expectation that the1st status in the response is SUCCESS, see Table 2.
- For Properties() invocations with the expectation that the1st status in the response is non-SUCCESS, see Table 3.
- For StartSession() invocations with the expectation that the1st status in the response is SUCCESS, see Table 4, plus make sure that upon successful invocation of StartSession using a C_PIN authority, the Tries column SHALL be reset to 0.
- For StartSession() invocations with the expectation that the1st status in the response is non-SUCCESS, see Table 5.
- For Get() invocations with the expectation that the1st status in the response is SUCCESS, see Table 7.
- For Next() or GetACL() invocations with the expectation that the1st status in the response is SUCCESS, see Table 8.
- For Set(), GenKey(), Activate(), Revert() or RevertSP() invocations with the expectation that the1st status in the response is SUCCESS, see Table 9.
- For Get(), Set(), Next() GetACL(), GenKey(), Activate(), Revert() or RevertSP() invocations with the expectation that the1st status in the response is non-SUCCESS, see Table 10.

- For Start Transaction requests, see Table 11.
- For End Transaction requests with the expectation of transaction commit, see Table 12.
- For End Transaction requests with the expectation of transaction abort, see Table 13.
- For End of Session requests with the expectation of successful session end, see Table 14.

Table 1 Properties Method Response (SUCCESS case with HostProperties)

Token Type		Value
Call (‘F8’)		(n/a)
Byte token		0x00 00 00 00 00 00 00 FF
Byte token		0x00 00 00 00 00 00 FF 01
StartList (‘F0’)		(n/a)
StartList (‘F0’)		(n/a)
Repeat this sequence for each property	StartName (‘F2)	(n/a)
	Uinteger, integer or byte token (depending on the content*)	<a property’s name>
	Uinteger, integer or byte token (depending on the content**)	<the property’s value>
	EndName (‘F3)	(n/a)
EndList (‘F1’)		(n/a)
StartName (‘F2’)		(n/a)
Uinteger token		0
StartList (‘F0’)		(n/a)
Repeat this sequence for each property	StartName (‘F2’)	(n/a)
	Uinteger, integer or byte token (depending on the content*)	<a property’s name>
	Uinteger, integer or byte token (depending on the content**)	<the property’s value>
	EndName (‘F3)	(n/a)
EndList (‘F1’)		(n/a)
EndName (‘F3)		(n/a)
EndList (‘F1’)		(n/a)
EndOfData (‘F9’)		(n/a)
StartList (‘F0’)		(n/a)
Uinteger token		0
Uinteger token		0
Uinteger token		0
Optional, could be multiple additional status codes	Uinteger token	Vendor-specific
EndList (‘F1’)		(n/a)

Token Type	Value
Notes: *: For at least the properties which are defined in the CoreSpec[1], this SHALL be byte token. **: For at least the properties which are defined in the CoreSpec[1], this SHALL be uinteger token.	

Table 2 Properties Method Response (SUCCESS case without HostProperties)

Token Type			Value
Call (‘F8’)			(n/a)
Byte token			0x00 00 00 00 00 00 00 FF
Byte token			0x00 00 00 00 00 00 FF 01
StartList (‘F0’)			(n/a)
StartList (‘F0’)			(n/a)
Repeat this sequence for each property		StartName (‘F2)	(n/a)
		Uinteger, integer or byte token (depending on the content*)	<a property’s name>
		Uinteger, integer or byte token (depending on the content**)	<the property’s value>
		EndName (‘F3)	(n/a)
EndList (‘F1’)			(n/a)
EndList (‘F1’)			(n/a)
EndOfData (‘F9’)			(n/a)
StartList (‘F0’)			(n/a)
EndOfData (‘F9’)			(n/a)
StartList (‘F0’)			(n/a)
Uinteger token			0
Uinteger token			0
Uinteger token			0
Optional, could be multiple additional status codes		Uinteger token	Vendor-specific
EndList (‘F1’)			(n/a)
Notes: *: For at least the properties which are defined in the CoreSpec[1], this SHALL be byte token. **: For at least the properties which are defined in the CoreSpec[1], this SHALL be uinteger token.			

Table 3 Properties Method Response (non-SUCCESS case)

Token Type			Value
Call (‘F8’)			(n/a)
Byte token			0x00 00 00 00 00 00 00 FF
Byte token			0x00 00 00 00 00 00 FF 01
StartList (‘F0’)			(n/a)
Since this is a non-success case,		StartList (‘F0’)	(n/a)
		StartName (‘F2)	(n/a)

Token Type			Value
the TPer is not required to return the Properties parameter.	Repeat this sequence for each property.	Uinteger, Integer or Byte token (depending on the content*)	<a property's name>
		Uinteger, Integer or Byte token (depending on the content**)	<the property's value>
		EndName ('F3)	(n/a)
		EndList ('F1')	(n/a)
EndList ('F1')			(n/a)
EndOfData ('F9')			(n/a)
StartList ('F0')			(n/a)
Uinteger token			non-0
Uinteger token			0
Uinteger token			0
Optional, could be multiple additional status codes		Uinteger token	Vendor-specific
EndList ('F1')			(n/a)
Notes: *: For at least the properties which are defined in the CoreSpec[1], this SHALL be byte token. **: For at least the properties which are defined in the CoreSpec[1], this SHALL be uinteger token.			

Table 4 SyncSession Method (SUCCESS case)

Token Type			Value
Call ('F8')			(n/a)
Byte token			0x00 00 00 00 00 00 00 FF
Byte token			0x00 00 00 00 00 00 FF 03
StartList ('F0')			(n/a)
Uinteger token			The number specified in HostSessionID in the StartSession().
Uinteger token			Greater than or equal to 0x10 00 and less than 0x01 00 00 00 00
EndList ('F1')			(n/a)
EndOfData ('F9')			(n/a)
StartList ('F0')			(n/a)
Uinteger token			0
Uinteger token			0
Uinteger token			0
Optional, could be multiple additional status codes		Uinteger token	Vendor-specific
EndList ('F1')			(n/a)

Table 5 SyncSession Method (non-SUCCESS case)

Token Type	Value
Call ('F8')	(n/a)

Token Type	Value
Byte token	0x00 00 00 00 00 00 00 FF
Byte token	0x00 00 00 00 00 00 FF 03
StartList (‘F0’)	(n/a)
Uinteger token	The number specified in HostSessionID in the StartSession().
Uinteger token	Greater than or equal to 0x10 00 and less than 0x01 00 00 00 00
EndList (‘F1’)	(n/a)
EndOfData (‘F9’)	(n/a)
StartList (‘F0’)	(n/a)
Uinteger token	non-0
Uinteger token	0
Uinteger token	0
Optional, could be multiple additional status codes: Uinteger token	Vendor-specific
EndList (‘F1’)	(n/a)

Table 6 CloseSession Method

Token Type	Value
Call (‘F8’)	(n/a)
Byte token	0x00 00 00 00 00 00 00 FF
Byte token	0x00 00 00 00 00 00 FF 06
StartList (‘F0’)	(n/a)
Uinteger token	The Host Session Number of aborted Session
Uinteger token	The SP Session Number of aborted Session
EndList (‘F1’)	(n/a)
EndOfData (‘F9’)	(n/a)
StartList (‘F0’)	(n/a)
Uinteger token	0
Uinteger token	0
Uinteger token	0
Optional, could be multiple additional status codes: Uinteger token	Vendor-specific
EndList (‘F1’)	(n/a)
Notes:	

Table 7 Get Method Response (SUCCESS case)

Token Type	Value
StartList (‘F0’)	(n/a)
Uinteger, integer, byte, StartList, EndList, StartName or EndName token depending on the contents of returned data*	

Token Type		Value
Repeat this for each token to indicate a table/column value. Could be none or more additional tokens	Uinteger, integer, byte, StartList, EndList, StartName or EndName token depending on the contents of returned data*	
EndList (‘F1’)		(n/a)
EndOfData (‘F9’)		(n/a)
StartList (‘F0’)		(n/a)
Uinteger token		0
Uinteger token		0
Uinteger token		0
Optional, could be multiple additional status codes	Uinteger token	Vendor-specific
EndList (‘F1’)		(n/a)
Notes: *: When non tokens are there, this row SHALL NOT be taken into account as a condition. But when there is/are some, each token SHALL be checked if its token type is appropriate to indicate the value of a table or a column of an object.		

Table 8 Next or GetACL Method Response (SUCCESS case)

Token Type		Value
StartList (‘F0’)		(n/a)
StartList (‘F0’)		(n/a)
Repeat this for each UID. Could be none or more additional tokens	Byte Token*	
EndList (‘F1’)		(n/a)
EndList (‘F1’)		(n/a)
EndOfData (‘F9’)		(n/a)
StartList (‘F0’)		(n/a)
Uinteger token		0
Uinteger token		0
Uinteger token		0
Optional, could be multiple additional status codes	Uinteger token	Vendor-specific
EndList (‘F1’)		(n/a)
Notes: *: When no corresponding tokens are there, this row SHALL NOT be taken into account as a condition		

Table 9 General Method Response (SUCCESS case)

Token Type	Value
StartList (‘F0’)	(n/a)
EndList (‘F1’)	(n/a)
EndOfData (‘F9’)	(n/a)
StartList (‘F0’)	(n/a)
Uinteger token	0

Token Type		Value
Uinteger token		0
Uinteger token		0
Optional, could be multiple additional status codes	Uinteger token	Vendor-specific
EndList (‘F1’)		(n/a)

Table 10 General Method Response (non-SUCCESS case)

Token Type		Value
StartList (‘F0’)		(n/a)
EndList (‘F1’)		(n/a)
EndOfData (‘F9’)		(n/a)
StartList (‘F0’)		(n/a)
Uinteger token		non-0
Uinteger token		0
Uinteger token		0
Optional, could be multiple additional status codes	Uinteger token	Vendor-specific
EndList (‘F1’)		(n/a)

Table 11 Start Transaction Request Response

Token Type	Value
StartTransaction (‘FB’)	(n/a)
Uinteger token	0

Table 12 End Transaction Request Response (Transaction Commit case)

Token Type	Value
EndTransaction (‘FC’)	(n/a)
Uinteger token	0

Table 13 End Transaction Request Response (Transaction Abort case)

Token Type	Value
EndTransaction (‘FC’)	(n/a)
Uinteger token	1

Table 14 End of Session Response

Token Type	Value
EndOfSession (‘FA’)	(n/a)

2.6. Session Abort as Alternative to Error Status

Session abort is an acceptable alternative to any TCG error status response in any test case. For every test case in this specification that specifies a TCG error status code in the TPer response, session abort SHALL also be an acceptable response.

2.7. TPer Capability Discovery and Test Order

Some tests in this document have more than one proper behavior depending on the TPer's capabilities. The following defines how to identify the TPer's capabilities:

- For ATA devices, IDENTIFY DEVICE data Word 119 bit 6 and the result of SET FEATURES with subcommand code 0xC3 indicate whether or not the TPer supports Sense Data Reporting, and IDENTIFY DEVICE data Word 120 bit 6 indicates whether or not it's enabled (see [5]),
  - if the Sense Data Reporting feature is supported, then the tests that depend on that feature SHALL be run both with the Sense Data Reporting feature disabled and enabled. This statement affects the following test cases:
    A1-1-1-1-1, A1-1-1-1-2, A1-1-3-1-3, A1-1-3-1-4, A1-1-3-1-5, A1-2-1-1-1, A1-2-3-2-2, A1-2-3-2-3,
    A2-1-1-1-2, A2-2-1-1-2,
    A3-1-1-1-1,
    A4-1-3-1-1,
    A10-3-2-1-1,
    D4-2-2-2-1, D4-2-2-2-1(2), D4-2-2-3-1(2), D4-2-2-4-1(2), D4-3-2-2-1,
    D4-3-2-2-1(2), D4-3-2-2-2, D4-3-2-3-1(2), D4-3-2-4-1(2),
    D5-1-2-2-2, D5-1-2-2-3, D5-1-2-3-2(2), D5-1-2-3-3, D5-1-2-3-3(2), D5-1-2-4-1, D5-1-2-4-2;
- For ATA devices, IDENTIFY DEVICE data Word 60-61 or Word 100-103 identifies the TPer's maximum number of sectors (MAX LBA + 1),
  - For devices that are are based on ACS-2 [5], if Word 69 bit 3 is set to 1, then Words 230-233 identify the maximum number of sectors (MAX LBA + 1). In this case, Words 100-103 MAY be truncated to 32 bits;
- 'LockingEnabled' bit in Level 0 Discovery Response data when the TPer is in its OFS indicates whether or not the TPer supports the Locking SP in its Manufactured state when the TPer is in its OFS;
- 'Base ComID' and 'Number of ComIDs' fields in Level 0 Discovery Response data identifies the TPer's static ComID(s);
- 'Range Crossing' bit in Level 0 Discovery Response data identifies whether or not the TPer supports Range Crossing READ and WRITE;
- Response data of an IF-RECV with its Security Protocol = 0 and its Security Protocol Specific = 0, and the result of Stack Reset request to determine whether or not the TPer supports Stack Reset;
- Returning Properties() invocation from the TPer identifies the TPer communication properties;
- C_PIN table content in the Admin SP identifies the MSID credential value of the TPer;
- AccessControl table content in the Admin SP identifies whether or not the TPer supports Revert() to the Admin SP and the Locking SP;
- Table table content in the Locking SP identifies the sizes of MBR table and DataStore table;
- Table table content in the Locking SP indicates whether the TPer supports K_AES_128 table or K_AES_256 table;
- Table table content in the Locking SP indicates whether or not the TPer supports RestrictedCommands table;

- Authority table content in the Locking SP identifies how many Admin authorities and User authorities the TPer supports;
- C_PIN table contents in the Admin SP and Locking SP indicate the values the TPer supports in the TryLimit column; and,
- LockingInfo table content in the Locking SP identifies how many Locking Ranges the TPer supports.

Assuming the TPer's capabilities are as described above, the tests in this document can be run in any order.

2.8. Document Structure

This section describes Test items defined in this document. In 'Detail' column, the test menu and the expected behaviors for each test are described. In 'Group', 'Subgroup', and 'Subsubgroup' columns, some title, topic, specific target, or/and common conditions of the test described on the right-hand-side 'Detail' cells are contained. There might be some notes in 'Memo' column to show informative information.

In some rows, the 'Subgroup' says "Effects of successful method invocation", and the 'Subsubgroup' and 'Detail' contains "(Description) .. ." In that case, the description in "(Description).." affects all the following rows until a new message appears in 'Subgroup' column.

3. Test Cases

3.1. Section A: Basic Grammar

3.1.1. A2. Security Protocol = 0 related

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
A2. Security Protocol = 0 related					
A2-1-1-1-1	TRUSTED-RECEIVE and TRUSTED RECEIVE DMA, Security Protocol Specific = 0x0000	Data Transfer		If the Transfer Length in a TRUSTED RECEIVE (as well as TRUSTED RECEIVE DMA) is set to greater than or equal to 1, then the command SHALL succeed at the ATA interface level with ST = 0x50 if there is no any other factor which leads to an error.	
A2-1-1-1-2				If the Transfer Length in a TRUSTED RECEIVE (as well as TRUSTED RECEIVE DMA) is set to 0, then the data transfer SHALL NOT occur and the command SHALL be aborted at the ATA interface level with: i) ST = 0x51 and ER = 0x04 or ii) ST has bit 1 set, Sense Key = ILLEGAL REQUEST and ASC/ASCQ = INVALID FIELD IN CDB if the TPer supports Sense Data Reporting feature set and it is enabled.	
A2-1-2-1-2		Data content when the command succeeds		Byte 6 and byte 7 in the payload of the TRUSTED RECEIVE SHALL (as well as TRUSTED RECEIVE DMA) indicate more than or equal to 2.	
A2-1-2-1-3(1)				Byte 8 in the payload of the TRUSTED RECEIVE (as well as TRUSTED RECEIVE DMA) SHALL be set to 0x00.	
A2-1-2-1-3(2)				Byte 9 in the payload of the TRUSTED RECEIVE (as well as TRUSTED RECEIVE DMA) SHALL be set to 0x01.	
A2-1-2-1-3(3)				Byte 10 in the payload of the TRUSTED RECEIVE (as well as TRUSTED RECEIVE DMA) SHALL be set to: i) 0x02 if the TPer supports Security Protocol 0x02, or, ii) 0x00 if the TPer does not support Security Protocol 0x02.	
A2-2-1-1-1	TRUSTED-RECEIVE and TRUSTED RECEIVE DMA, Security Protocol Specific = 0x0001	Data Transfer		If the Transfer Length in a TRUSTED RECEIVE (as well as TRUSTED RECEIVE DMA) is set to greater than or equal to 1, then the command SHALL NOT fail at the ATA interface level because of that.	
A2-2-1-1-2				If the Transfer Length in a TRUSTED RECEIVE (as well as TRUSTED RECEIVE DMA) is set to 0, then the data transfer SHALL NOT occur and the command SHALL be aborted at the ATA interface level with: i) ST = 0x51 and ER = 0x04 or ii) ST has bit 1 set, Sense Key = ILLEGAL REQUEST and ASC/ASCQ = INVALID FIELD IN CDB if the TPer supports Sense Data Reporting feature set and it is enabled.	

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
A2-2-2-1-2		Data content when the command succeeds		Byte 2 and byte 3 in the payload of the TRUSTED RECEIVE SHALL (as well as TRUSTED RECEIVE DMA) be set to: i) 0x00 0x00 if the device does not have a certificate data in the following field which starts at Byte 4. ii) the length in byte of the certificate data which starts at byte 4.	

3.1.2. A3. Level 0 Discovery - Basic Grammar -

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
A3. Level 0 Discovery - Basic Grammar -					
A3-1-1-1-1	Level 0 Discovery	Data Transfer		If the Transfer Length in a TRUSTED RECEIVE (as well as TRUSTED RECEIVE DMA) is set to 0, the Security Protocol is set to 1 and the Security Protocol Specific is set to 0x0001, then the data transfer SHALL NOT occur and the command SHALL be aborted at the ATA interface level with: i) ST = 0x51 and ER = 0x04 or ii) ST has bit 1 set, Sense Key = ILLEGAL REQUEST and ASC/ASCQ = INVALID FIELD IN CDB if the TPer supports Sense Data Reporting feature set and it is enabled.	
A3-1-1-1-2				If the Transfer Length in a TRUSTED RECEIVE (as well as TRUSTED RECEIVE DMA) is set to greater than or equal to 1, the Security Protocol is set to 1 and the Security Protocol Specific is set to 0x0001, then the command SHALL NOT fail at the ATA interface level because of that.	

3.1.3. A4. Synchronous Communication Protocol

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
A4. Synchronous Communication Protocol					
A4-1-1-1-1	TRUSTED SEND and TRUSTED SEND DMA	The TPer is in the Awaiting IF-SEND state for at least one of the static ComIDs	Static ComIDs	Immediately after a power-on reset, the TPer SHALL be in the Awaiting IF-SEND state for its static ComIDs, which means if the TPer has just processed a power-on reset, the Transfer Length in a TRUSTED SEND (as well as TRUSTED SEND DMA) is set to any value between 1 and the value the TPer's MaxComPacketSize property for the ComID in the command parameter indicates, the Security Protocol is set to 1, and the Security Protocol Specific is set to one of the static ComIDs, then the command SHALL succeed at the ATA interface level with ST = 0x50 if there is no any other factor which leads to an error.	
A4-1-1-1-3				If the TPer has just transitioned from the Awaiting IF-RECV state to the Awaiting IF-SEND state for one of the TPer's static ComIDs, the Transfer Length in a TRUSTED SEND (as well as TRUSTED SEND DMA) is set to any value between 1 and the value the TPer's MaxComPacketSize property for the ComID in the command parameter indicates, the Security Protocol is set to 1, and the Security Protocol Specific is set to the state-changed ComID, then the command SHALL succeed at the ATA interface level with ST = 0x50 if there is no any other factor which leads to an error.	
A4-1-3-1-1		The TPer is in the Awaiting IF-RECV state for at least one of the static ComIDs	Static ComIDs	If the ComID in a TRUSTED SEND (as well as TRUSTED SEND DMA) is set to a ComID that is in the Awaiting IF-RECV state, the Security Protocol is set to 1, and the Transfer Length is set to any value between 1 and the value the TPer's MaxComPacketSize property for the ComID in the command parameter indicates, then the data transfer SHALL NOT occur and the command SHALL be aborted at the ATA interface level with: i) ST = 0x51 and ER = 0x04 or ii) ST has bit 1 set, Sense Key = ILLEGAL REQUEST and ASC/ASCQ = COMMAND SEQUENCE ERROR if the TPer supports Sense Data Reporting feature set and it is enabled. The TPer SHALL stay in the Awaiting IF-RECV state for that ComID.	
A4-2-1-2-1	TRUSTED RECEIVE and TRUSTED RECEIVE DMA	The TPer is in the Awaiting IF-SEND state for at least one of the static ComIDs	ComID for Level 0 Discovery	If the Security Protocol Specific in a TRUSTED RECEIVE (as well as TRUSTED RECEIVE DMA) is set to 0x0001, the Security Protocol is set to 1, and the Transfer Length is set to greater than or equal to 1, then the command SHALL succeed at the ATA interface level with ST = 0x50, the proper contents SHALL be returned.	

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
A4-2-1-2-3			Static ComIDs	If the Security Protocol Specific in a TRUSTED RECEIVE (as well as TRUSTED RECEIVE DMA) is set to a ComID that is in the Awaiting IF-SEND state, the Security Protocol is set to 1 and the requested Transfer Length is set to greater than or equal to 1, then the command SHALL succeed at the ATA interface level with ST = 0x50 and the contents of "All Response(s) returned, no further data" (OutstandingData = 0, MinTransfer = 0 and Length = 0) with pad bytes of 0x0 as needed to meet the requested Transfer Length SHALL be returned.	
A4-2-3-2-1		The TPer is in the Awaiting IF-RECV state for at least one of the static ComIDs	ComID for Level 0 Discovery	If the Security Protocol Specific in a TRUSTED RECEIVE (as well as TRUSTED RECEIVE DMA) is set to 0x0001, the Security Protocol is set to 1 and the Transfer Length is set to greater than or equal to 1, then the command SHALL succeed at the ATA interface level with ST = 0x50, the proper contents SHALL be returned.	
A4-2-3-2-2			Static ComIDs	If the Security Protocol Specific in a TRUSTED RECEIVE (as well as TRUSTED RECEIVE DMA) is set to a ComID that is in the Awaiting IF-RECV state, the Security Protocol is set to 1, the Transfer Length is set to greater than or equal to 1 and is sufficient to get all the data the TPer has prepared, and no further response data is to come, then the command SHALL succeed at the ATA interface level with ST = 0x50 and the contents of "All Response(s) returned, no further data" (OutstandingData = 0, MinTransfer = 0 and Length = <data length>) SHALL be returned. And the TPer SHALL stay in or immediately transition to the Awaiting IF-SEND state for that ComID.	
A4-2-3-2-3				If the Security Protocol Specific in a TRUSTED RECEIVE (as well as TRUSTED RECEIVE DMA) is set to a ComID that is in the Awaiting IF-RECV state, the Security Protocol is set to 1, and the Transfer Length is set to greater than or equal to 1 but is insufficient to get any data the TPer has prepared, then the command SHALL succeed at the ATA interface level with ST = 0x50 and the contents of "Response ready, insufficient transfer length requested" (OutstandingData = <total data available>, MinTransfer = <minimum request required to transfer a packet> and Length = 0) SHALL be returned. And the TPer SHALL stay in or immediately transition to the Awaiting IF-RECV state for that ComID.	

3.1.4. A5. ComPacket/Packet/Subpacket

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
A5. ComPacket/Packet/Subpacket					
A5-1-1-1-2	ComPacket header in IF-SEND	Reserved field		If this field is not set to all-0s, then the ComPacket SHALL NOT be discarded and the method(s) in the ComPacket, if any, SHALL NOT fail because of that.	
A5-1-2-2-2		ExtendedComID	ComID part Invalid	If this field is not set to the same number as the ComID of the command parameter in the TRUSTED SEND (or TRUSTED SEND DMA), then the entire data SHALL be transferred successfully (if no error factor), the status field of the ATA interface SHALL be set to 0x50 (if no error factor), but the data SHALL be discarded by the TPer, and the TPer SHALL stay in or immediately transition to the Awaiting IF-SEND state for the static ComID.	
A5-1-2-5-1			ComID Extension part Invalid	If this field is not set to all-0s, then the entire data SHALL be transferred successfully (if no error factor), the status field of the ATA interface SHALL be set to 0x50 (if no error factor), but the data SHALL be discarded by the TPer, and the TPer SHALL stay in or immediately transition to the Awaiting IF-SEND state for the static ComID.	
A5-1-3-1-2		OutstandingData field		If this field is not set to all-0s, then the ComPacket SHALL NOT be discarded and the method(s) in the ComPacket, if any, SHALL NOT fail because of that.	
A5-1-4-1-2		MinTransfer field		If this field is not set to all-0s, then the ComPacket SHALL NOT be discarded and the method(s) in the ComPacket, if any, SHALL NOT fail because of that.	
A5-1-5-1-2		Length field		If this field is set to a value that indicates an amount of data that does not fit within the data transferred by the TRUSTED SEND (or TRUSTED SEND DMA), then the data transfer SHALL have been completed, and the command SHALL succeed with ST = 0x50 (if no error factor), but the data SHALL be discarded by the TPer, and the TPer SHALL stay in or immediately transition to the Awaiting IF-SEND state for that associated ComID.	
A5-1-5-1-2(2)				If this field is set to a value that is less than the length of a Packet header field, 24, then the ComPacket SHALL be discarded by the TPer, and the TPer SHALL stay in or immediately transition to the Awaiting IF-SEND state for that associated ComID.	
A5-1-5-1-3				If this field is set to a value that indicates there are some bytes that are not contained in the ComPacket in the data transferred, then the bytes are supposed to be padding bytes. The padding bytes should be set to all-0s, however the bytes being not set to all-0s SHALL be ignored.	

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
A5-2-1-1-2	Packet header in IF-SEND	Session field		If this field is not set to one of the open sessions' numbers, then the Packet SHALL be discarded by the TPer. (* The discarding of the Packet in the TPer could be verified by seeing that the TPer does not return back a Packet of which session number is the same as the Packet supposed to be discarded, but eventually returns a ComPacket of which Outstanding field is set to 0.)	
A5-2-3-1-2		Reserved field		If this field is not set to all-0s, then the Packet SHALL NOT be discarded, the session SHALL NOT be aborted and the method(s) in the Packet, if any, SHALL NOT fail because of that.	
A5-2-6-1-2		Length field		If this field is set to a value that indicates an amount of data that does not fit within the ComPacket payload, then: i) if the packet is for a Regular Session, the session SHALL be aborted. ii) if the packet is for the Control Session, the packet SHALL be discarded by the TPer.	
A5-2-6-1-2(2)				If this field is set to a value that is less than the length of a Subpacket header field, 12, then: i) if the packet is for a Regular Session, the session SHALL be aborted. ii) if the packet is for the Control Session, the packet SHALL be discarded by the TPer.	
A5-3-1-1-1(1)				(* If the start point of the header field of this Subpacket is not aligned with 4-byte boundary in the TRUSTED SEND (or TRUSTED SEND DMA) command payload, the Subpacket MAY not be parsed by the TPer correctly.)	
A5-3-1-1-2	Subpacket header in IF-SEND	Reserved field		If this field is not set to all-0s, then the Subpacket SHALL NOT be discarded, the session SHALL NOT be aborted and the method(s) in the Subpacket, if any, SHALL NOT fail because of that.	
A5-3-3-1-2		Length field		If this field is set to a value that indicates an amount of data exceeds the Packet, then: i) if the packet is for a Regular Session, the session SHALL be aborted. ii) if the packet is for the Control Session, the packet SHALL be discarded by the TPer.	

3.1.5. A6. Method Invocation/Response - Basic Grammar -

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
A6. Method Invocation/Response -Basic Grammar-					
A6-0-1-1-1	Method invocation in general	Variety of Atom Size		Any atom within the token stream of the method Invocation could be encoded with any size of atom (Short atom, Medium atom, etc.). The method invocation SHALL NOT fail because of that differentiation.	
A6-1-1-1-1(1)	Method invocation in a Regular Session	Invalid token		If invalid token was used, the method call SHALL lead to session abort.	
A6-1-1-2-1		InvokingID	Nonexistent UID	If requested InvokingID is nonexistent in the SP, the method call SHALL fail with NOT_AUTHORIZED, unless the MethodID is 'Get', in which case an empty method response with SUCCESS status SHALL be returned.	
A6-1-1-3-1(2)			Unexpected token	If unexpected token was used, the method call SHALL fail with NOT_AUTHORIZED.	
A6-1-2-2-1		MethodID	Nonexistent UID	Requested MethodID is nonexistent in the SP, the method call SHALL fail with NOT_AUTHORIZED.	
A6-1-2-3-1(2)			Unexpected token	If unexpected token was used, the method call SHALL fail with NOT_AUTHORIZED.	
A6-1-3-1-1		InvokingID/MethodID Not Authorized		If the InvokingID/MethodID combination exists in AccessControl table, but no ACE in the ACL is satisfied by the currently authenticated Authorities, the method call SHALL fail with NOT_AUTHORIZED, unless the MethodID is Get, in which.case an empty method response with SUCCESS status SHALL be returned	
A6-1-3-1-1(2)				If the InvokingID and MethodID exist, but the combination of those does not exist in AccessControl table, the method call SHALL fail with NOT_AUTHORIZED, unless the MethodID is Get, in which.case an empty method response with SUCCESS status SHALL be returned.	
A6-1-4-2-1		Parameters start (StartList token)	Unexpected token	If the beginning of parameter list is not 'F0' token, the method call SHALL lead to session abort.	
A6-1-4-2-1(1)		Within Parameters	Unexpected token	If there is any Unexpected token encoded inside the Parameters field, the method SHALL fail with INVALID_PARAMETER.	
A6-1-4-2-1(2)			Same optional parameter encoding	If one or more optional parameter(s) are (M) for a method in Opal SSC[3] and one of those parameters is encoded twice or more in that method invocation, the method invocation SHALL fail with INVALID_PARAMETER.	
A6-1-4-2-1(3)			Non-ascending order of optional parameter encoding	If more than one optional parameter are (M) for a method in Opal SSC[3] and two of those parameters are encoded in descending order, the method SHALL fail with INVALID_PARAMETER.	

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
A6-1-5-2-1		Parameters end (EndList token)	Unexpected token	If the end of parameter list is not 'F1' token, the method call SHALL lead to session abort.	
A6-1-6-2-1		End of Data token	Unexpected token type	If the parameter list is not followed by 'F9' token, the method call SHALL lead to session abort.	
A6-1-7-2-1		Status code start (StartList token)	Unexpected token type	If the beginning of status list is not 'F0' token, the method SHALL lead to session abort.	
A6-1-8-1-2		Status code	1st status token (=0x00)	If a bigger size of atom than necessary (a short, medium or long atom) is used for 1st status, the method call SHALL NOT fail by that.	
A6-1-8-2-1			1st status code (!=0x00)	If the value of 1st status code is not 0x00, the method call SHALL fail (no effects of the method call SHALL happen). The returned 1st status from the TPer to the host SHALL be the status code that the host sent during the method invocation.	
A6-1-8-3-2			2nd and 3rd status codes	If the 2nd and 3rd status codes are not all-0x00s, the method SHALL NOT fail by that.	
A6-1-8-6-1			Unexpected token	If any status code is encoded by a non-uinteger atom (for example an integer atoms or a byte atom) when the method was invoked within a Regular Session, the method call SHALL lead to session abort.	
A6-1-9-2-1		Status code end (EndList token)	Unexpected token	If the end of status list is not 'F1' token, the method call SHALL lead to session abort.	
A6-3-1-2-1	Method invocation in a Control Session	InvokingID	Invalid UID	If requested InvokingID is not set to 0x00 00 00 00 00 00 00 FF, the method call SHALL be ignored by the TPer and no response for that method SHALL be prepared.	
A6-3-1-3-1(2)			Unexpected token	If unexpected token was used, the method call SHALL be ignored by the TPer and no response for that method SHALL be prepared	
A6-3-2-2-1		MethodID	Nonexistent UID	Requested MethodID is not set to any MethodID of the methods of Session Manager defined in the CoreSpec[1], the method call SHALL be ignored by the TPer and no response for that method SHALL be prepared.	
A6-3-2-3-1(2)			Unexpected token	If invalid token or unexpected was used, the method call SHALL be ignored by the TPer and no response for that method SHALL be prepared	
A6-3-4-2-1		Parameters start (StartList token)	Unexpected token	If the beginning of parameter list is not 'F0' token, the method call SHALL be ignored by the TPer and no response for that method SHALL be prepared.	
A6-3-4-2-1(1)		Within Parameters	Unexpected token	If there is any Unexpected token encoded inside the Parameters field, the method SHALL: i) be ignored by the TPer and no response for that method SHALL be prepared, or, ii) fail with INVALID_PARAMETER.	

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
A6-3-4-2-1(2)			Same optional parameter encoding	If one or more optional parameter(s) are (M) for a method in Opal SSC[3] and one of those parameters is encoded twice or more in that method invocation, the method invocation SHALL fail with INVALID_PARAMETER.	
A6-3-4-2-1(3)			Non-ascending order of optional parameter encoding	If more than one optional parameter are (M) for a method in Opal SSC[3] and two of those parameters are encoded in descending order, the method SHALL fail with INVALID_PARAMETER.	
A6-3-5-2-1		Parameters end (EndList token)	Unexpected token	If the end of parameter list is not 'F1' token, the method call SHALL be ignored by the TPer and no response for that method SHALL be prepared.	
A6-3-6-2-1		End of Data token	Unexpected token	If the parameter list is not followed by 'F9' token, the method call SHALL be ignored by the TPer and no response for that method SHALL be prepared.	
A6-3-7-2-1		Status code start (StartList token)	Unexpected token	If the beginning of status list is not 'F0' token, the method SHALL be ignored by the TPer and no response for that method SHALL be prepared.	
A6-3-8-1-2		Status code	1st status token (=0x00)	If a bigger size of atom than necessary (a short, medium or long atom) is used for 1st status, the method call SHALL NOT fail by that.	
A6-3-8-2-1			1st status code (!=0x00)	If the value of 1st status code is not 0x00, the method call SHALL fail (no effects of the method call SHALL happen). The returned 1st status from the TPer to the host SHALL be the status code that the host sent during the method invocation.	
A6-3-8-3-2			2nd and 3rd status codes	If the 2nd and 3rd status codes are not all-0x00, the method SHALL NOT fail by that.	
A6-3-8-6-1			Unexpected token	If any status code is encoded by a non-uinteger atom (for example an integer atoms or a byte atom) when the method was invoked, the method call SHALL be ignored by the TPer and no response for that method SHALL be prepared.	
A6-3-9-2-1		Status code end (EndList token)	Unexpected token	If the end of status list is not 'F1' token, the method call SHALL be ignored by the TPer and no response for that method SHALL be prepared.	

3.1.6. A7. Transaction

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
A7. Transaction					
A7-1-1-2-1(2)	Transaction start/commit/abort in a Regular Session	Transaction Start Grammar	Status code	If the status code after a Start Transaction token indicates any number and is encoded with any size of uinteger atom, a tiny or short or medium or long atom, the Transaction start request SHALL NOT fail and the session SHALL NOT be aborted because of that. The TPer SHALL mirror a Start Transaction token and a following status code of which value is zero in the corresponding place in the message stream.	
A7-1-1-2-5				If the status code is encoded with a non-uinteger atom (for example an integer atom or a byte atom), the session SHALL be aborted.	
A7-1-1-2-6				If the status code is not encoded, the session SHALL be aborted.	
A7-1-2-2-1(1)		Transaction End Grammar	Status code	If the status code after an End Transaction token indicates 0 and is encoded with any size of uinteger atom, a tiny or short or medium or long atom, the Transaction commit request SHALL NOT fail and the session SHALL NOT be aborted because of that. The TPer SHALL mirror an End Transaction token and a following status code of which value is zero in the corresponding place in the message stream.	
A7-1-2-2-1(2)				If the status code after an End Transaction token indicates non-0 and is encoded with any size of uinteger atom, a tiny or short or medium or long atom, the Transaction abort request SHALL NOT fail and the session SHALL NOT be aborted because of that. The TPer SHALL mirror an End Transaction token and a following status code of which value is one in the corresponding place in the message stream.	
A7-1-2-2-5				If the status code is encoded with a non-uinteger atom (for example an integer atom or a byte atom), the session SHALL be aborted.	
A7-1-2-2-6				If the status code is not encoded, the session SHALL be aborted.	
A7-1-3-1-1		Transaction Start/End situation	Transaction start attempt	If the (Start Transaction token) + (any status code) is sent and that does not cause the number of the open transactions in the session to exceed the value of the TPer's MaxTransactionLimit property, the Transaction start request SHALL NOT fail and the session SHALL NOT be aborted because of that.	
A7-1-3-1-2				If the (Start Transaction token) + (any status code) is sent and that causes the number of the open transactions in the session to exceed the value of the TPer's MaxTransactionLimit property, the session SHALL be aborted.	
A7-1-3-2-1			Transaction commit attempt	If the (End Transaction token) + (status code 0) is sent outside of a transaction, the session SHALL be aborted.	

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
A7-1-3-2-2				If the (End Transaction token) + (status code 0) is sent within a transaction, and: i) if the Transaction commit does not fail, the Transaction commit request SHALL be completed and (End Transaction token) + (status code 0) SHALL be sent in the equivalent place the host encoded in the message stream. ii) if the Transaction commit fails, the Transaction SHALL be aborted and (End Transaction token) + (status code 1) SHALL be sent in the equivalent place the host encoded in the message stream.	
A7-1-3-3-1			Transaction abort attempt	If the (End Transaction token) + (status code non-0) is sent outside of a transaction, the session SHALL be aborted.	
A7-1-3-3-2				If the (EndTransaction token) + (status code non-0) is sent within a transaction, the Transaction abort request SHALL NOT fail and (End Transaction token) + (status code 1) SHALL be sent in the equivalent place the host encoded in the message stream.	
A7-1-3-4-1			Stand-alone Transaction Start	If there are only Start Transaction token and a status token after that in the Subpacket, the Transaction start request SHALL NOT fail and the session SHALL NOT be aborted because of that.	
A7-1-3-5-1			Stand-alone Transaction End	When a transaction is in process,and if a Subpacket is sent where only one End Transaction token and one status token after that are contained in the Subpacket, the Transaction end request SHALL NOT cause a session abort, and the End Transaction reply from the TPer SHALL contain the proper status (either Commit or Abort) based on the status of the End Transaction sent by the host, unless there is some other error factor.	
A7-1-3-6-1			Multiple transaction start/end	If one or more Transaction commit(s)/abort(s) occur within the session, the next time of Transaction start request SHALL NOT fail because of that.	
A7-1-3-8-1			Transaction attempt in a Control Session	If a "StartTransaction token + status" or an "EndTransaction token + status" is encoded outside of a method invocation in a Control Session, the token SHALL be discarded.	
A7-1-6-1-1		Transaction + Session abort		If a transaction has been started successfully and the session gets aborted without a transaction commit or abort request to the transaction, the transaction SHALL be aborted.	
A7-1-7-1-1		Transaction + Session close		If a transaction has been started successfully and the session gets closed (accomplished by sending an EOS token.) without a transaction commit or abort request to the transaction, the transaction SHALL be aborted.	
A7-1-8-1-1		Transaction + Power Cycle		If a transaction has been started successfully and a power cycle happens without a transaction commit or abort request to the transaction, the transaction SHALL be aborted.	

3.1.7. A8. Ending Session

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
A8. Ending Session					
A8-1-1-1-1	End of Session token	Grammar		If an End of Session token, '0xFA', is sent in a Regular session, the TPer SHALL close the session and return the response prepared before closing the session. That should include the responses to the methods that were invoked prior to the EOS token. The TPer SHALL put the End of Session token at same position as the host put.	
A8-1-1-1-1(2)				If an End of Session token is encoded between method invocations while a transaction is in progress in a Regular Session, the TPer SHALL close the session and return the response prepared before closing the session. That should include the responses to the methods that were invoked prior to the EOS token. The TPer SHALL put the End of Session token at same position as the host put.	(Note: transaction SHALL NOT committed if the End of Session token was encoded before an "End Transaction token + status" while a transaction was in progress, because the session was ended before committing the transaction.)
A8-1-1-1-1(3)				If an End of Session token is encoded outside of a method invocation in a Control Session, the token SHALL be discarded.	
A8-1-2-1-1		Effect of successful End of Session		If the TPer sends an End of Session token in a Regular session via TRUSTED RECEIVE, the session SHALL be closed immediately after that.	
A8-1-2-1-2				If there are some tokens which follows the End of Session token in a Regular session, the tokens SHALL be discarded by the TPer.	
A8-1-4-1-1		Session after a session closing		If a session is closed by exchanging End of Session tokens and a new session startup method was invoked, the invocation SHALL NOT fail because of that.	
A8-2-2-10-1	CloseSession (* This is only tested if the TPer supports CloseSession.)	CloseSession from TPer - Effect-		If a CloseSession was sent by the TPer, verify that the session was aborted.	
A8-2-3-1-1		Session after a CloseSession		If a session is aborted and then a new session startup method was invoked, the invocation SHALL NOT fail because of that.	
A8-3-2-1-1	Session Timeout (These conditions can only be tested if there is a finite timeout.)	Timing of Session timeout		If the TPer has reached its MaxSessions limit, and the session timeout value has been exceeded on at least one session, then opening a new session SHALL succeed (if there is no other error in the session startup).	
A8-3-4-1-1		Session after a session timeout		If a session is aborted due to the timeout and then a new session startup method was invoked, the invocation SHALL NOT fail because of that.	

3.1.8. A9. Empty Atom

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
A9. Empty Atom					
A9-1-1-1-1	Empty Atom	Empty atoms between any tokens	Just before a Call token, '0xF8'	The existence of Empty atoms, 0xFF, at the position described in the left cell SHALL NOT lead to an error such as session abort or Subpacket discarding, etc. (The TPer SHALL NOT change the behavior based on the existence of the Empty Atoms.)	
A9-1-1-2-1			Between a Call token, 0xF8, and an 'InvokingID'	The existence of Empty atoms, 0xFF, at the position described in the left cell SHALL NOT lead to an error such as session abort or Subpacket discarding, etc. (The TPer SHALL NOT change the behavior based on the existence of the Empty Atoms.)	
A9-1-1-3-1			Between an 'InvokingID' and a 'MethodID'	The existence of Empty atoms, 0xFF, at the position described in the left cell SHALL NOT lead to an error such as session abort or Subpacket discarding, etc. (The TPer SHALL NOT change the behavior based on the existence of the Empty Atoms.)	
A9-1-1-4-1			Between a MethodID and Parameters' beginning, '0xF0'	The existence of Empty atoms, 0xFF, at the position described in the left cell SHALL NOT lead to an error such as session abort or Subpacket discarding, etc. (The TPer SHALL NOT change the behavior based on the existence of the Empty Atoms.)	
A9-1-1-5-1			Among tokens inside of a Parameters	The existence of Empty atoms, 0xFF, at the position described in the left cell SHALL NOT lead to an error such as session abort or Subpacket discarding, etc. (The TPer SHALL NOT change the behavior based on the existence of the Empty Atoms.)	
A9-1-1-6-1			Between Parameters' ending, '0xF1', and an End of Data token, '0xF9'	The existence of Empty atoms, 0xFF, at the position described in the left cell SHALL NOT lead to an error such as session abort or Subpacket discarding, etc. (The TPer SHALL NOT change the behavior based on the existence of the Empty Atoms.)	
A9-1-1-7-1			Between End of Data token, '0xF9', and Status Code List's beginning, '0xF0'	The existence of Empty atoms, 0xFF, at the position described in the left cell SHALL NOT lead to an error such as session abort or Subpacket discarding, etc. (The TPer SHALL NOT change the behavior based on the existence of the Empty Atoms.)	
A9-1-1-8-1			Among tokens inside of a Status Code List	The existence of Empty atoms, 0xFF, at the position described in the left cell SHALL NOT lead to an error such as session abort or Subpacket discarding, etc. (The TPer SHALL NOT change the behavior based on the existence of the Empty Atoms.)	
A9-1-1-9-1			Just after Status Code List's ending	The existence of Empty atoms, 0xFF, at the position described in the left cell SHALL NOT lead to an error such as session abort or Subpacket discarding, etc. (The TPer SHALL NOT change the behavior based on the existence of the Empty Atoms.)	

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
A9-1-1-10-1			Just before a Transaction Start token, '0xFB'	The existence of Empty atoms, 0xFF, at the position described in the left cell SHALL NOT lead to an error such as session abort or Subpacket discarding, etc. (The TPer SHALL NOT change the behavior based on the existence of the Empty Atoms.)	
A9-1-1-11-1			Between a Transaction Start token, '0xFB', and the status code	The existence of Empty atoms, 0xFF, at the position described in the left cell SHALL NOT lead to an error such as session abort or Subpacket discarding, etc. (The TPer SHALL NOT change the behavior based on the existence of the Empty Atoms.)	
A9-1-1-12-1			Just after the status after a Transaction Start token, '0xFB'	The existence of Empty atoms, 0xFF, at the position described in the left cell SHALL NOT lead to an error such as session abort or Subpacket discarding, etc. (The TPer SHALL NOT change the behavior based on the existence of the Empty Atoms.)	
A9-1-1-13-1			Just before a Transaction End token, '0xFC'	The existence of Empty atoms, 0xFF, at the position described in the left cell SHALL NOT lead to an error such as session abort or Subpacket discarding, etc. (The TPer SHALL NOT change the behavior based on the existence of the Empty Atoms.)	
A9-1-1-14-1			Between a Transaction End token, '0xFC', and the status code	The existence of Empty atoms, 0xFF, at the position described in the left cell SHALL NOT lead to an error such as session abort or Subpacket discarding, etc. (The TPer SHALL NOT change the behavior based on the existence of the Empty Atoms.)	
A9-1-1-15-1			Just after the status after a Transaction End token, '0xFC'	The existence of Empty atoms, 0xFF, at the position described in the left cell SHALL NOT lead to an error such as session abort or Subpacket discarding, etc. (The TPer SHALL NOT change the behavior based on the existence of the Empty Atoms.)	
A9-1-2-1-1		Empty atoms in plural places		If empty atoms are encoded in every possible positions in a method invocation or/and a transaction commit or abort request or/and an End of Session token, the method/request SHALL NOT fail because of that.	

3.1.9. A10. Properties()

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
A10. Properties()					
A10-1-6-2-6	Properties() invocation -Grammar-	Parameters – Optional Parameter -	HostProperties parameter	If the name of a Name-Value pair in the list of 'HostProperties' parameter's value indicates a name the TPer does not support as a property, the Name-Value pair SHALL be ignored, but the method invocation SHALL NOT fail because of that.	
A10-1-6-2-7(2)				If the 'HostProperties' parameter is encoded in the method invocation, the method response SHALL contain the ‘HostProperties’ parameter which contains all the host's properties the Opal SSC[3] requires.	
A10-1-6-3-1			HostProperties parameter -MaxComPacketSize-	If the name of a Name-Value pair in the list of 'HostProperties' parameter's value is "MaxComPacketSize", an ASCII text (excluding the double quotation marks ("... ")), and the value is set to a number that is less than or equal to 0x800, the method response SHALL contain the Name-Value pair of which name is "MaxComPacketSize" and value is 0x800 in the 'HostProperties' parameter.	
A10-1-6-5-1			HostProperties parameter -MaxPacketSize-	If the name of a Name-Value pair in the list of 'HostProperties' parameter's value is "MaxPacketSize", an ASCII text (excluding the double quotation marks ("... ")), and the value is set to a number that is less than or equal to 0x7EC, the method response SHALL contain the Name-Value pair of which name is "MaxPacketSize" and value is 0x7EC in the 'HostProperties' parameter.	
A10-1-6-6-1			HostProperties parameter -MaxIndTokenSize-	If the name of a Name-Value pair in the list of 'HostProperties' parameter's value is "MaxIndTokenSize", an ASCII text (excluding the double quotation marks ("... ")), and the value is set to a number that is less than or equal to 0x7C8, the method response SHALL contain the Name-Value pair of which name is "MaxIndTokenSize" and value is 0x7C8 in the 'HostProperties' parameter.	
A10-1-6-7-1			HostProperties parameter -MaxPackets-	If the name of a Name-Value pair in the list of 'HostProperties' parameter's value is "MaxPackets", an ASCII text (excluding the double quotation marks ("... ")), and the value is set to a number, the method response SHALL contain the Name-Value pair of which name is "MaxPackets" and value is the equal or a lesser value sent by the host or the maximum value supported by the TPer for the host's MaxPackets property.	

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
A10-1-6-8-1			HostProperties parameter -MaxSubpackets-	If the name of a Name-Value pair in the list of 'HostProperties' parameter's value is "MaxSubpackets", an ASCII text (excluding the double quotation marks ("... ")), and the value is set to a number, the method response SHALL contain the Name-Value pair of which name is "MaxSubpackets" and value is the equal or a lesser value sent by the host or the maximum value supported by the TPer for the host's MaxSubpackets property.	
A10-1-6-9-1			HostProperties parameter -MaxMethods-	If the name of a Name-Value pair in the list of 'HostProperties' parameter's value is "MaxMethods", an ASCII text (excluding the double quotation marks ("... ")), and the value is set to a number, the method response SHALL contain the Name-Value pair of which name is "MaxMethods" and value is the equal or a lesser value sent by the host or the maximum value supported by the TPer for the host's MaxMethods property.	
A10-1-6-15-1			Omission of HostProperties	If 'HostProperties' parameter' is not encoded in the method invocation, the method response SHALL NOT contain the 'HostProperties' parameter.	
A10-3-1-1-2	Properties effect	HostProperties	MaxComPacketSize	If the host encodes a method invocation in A Packet/Subpacket with no ST, ET, or EOS control tokens, and the TPer’s response would contain more data than can fit in a ComPacket based on the host’s MaxComPacketSize property, the method invocation SHALL fail with an empty results list and a status of RESPONSE_OVERFLOW.	
A10-3-1-3-2			MaxPacketSize	If the host encodes a method invocation in a Packet/Subpacket with no ST, ET, or EOS control tokens, and the TPer’s response would contain more data than can fit in a Packet based on the host’s MaxPacketSize property, the method invocation SHALL fail with an empty results list and a status of RESPONSE_OVERFLOW.	
A10-3-1-4-2			MaxIndTokenSize	If the TPer is unable to send a response to the host using tokens all of which are less than or equal to the host's MaxIndTokenSize property, i) if the Packet is for a Regular Session, then the session SHALL be aborted. ii) if the Packet is for the Control Session, this scenario cannot be tested.	

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
A10-3-2-1-1		TPerProperties	MaxComPacketSize	If a host sends a ComPacket larger than the TPer's MaxComPacketSize property value, the command SHALL be aborted at the ATA interface level with: i) ST = 0x51 and ER = 0x04 or, ii) ST has bit 1 set, Sense Key = ILLEGAL REQUEST and ASC/ASCQ = INVALID FIELD IN CDB if the TPer supports Sense Data Reporting feature set and it is enabled.	
A10-3-2-3-1			MaxPacketSize	If a host sends a Packet larger than the TPer's MaxPacketSize property value, i) if the Packet is for a Regular Session, the session SHALL be aborted. ii) if the Packet is for the Control Session, the Packet SHALL be discarded by the TPer.	
A10-3-2-4-1			MaxIndTokenSize	If a host sends a token larger than the TPer's MaxIndTokenSize property value, i) if the Packet is for a Regular Session, the session SHALL be aborted. ii) if the Packet is for the Control Session, the Packet SHALL be discarded by the TPer.	
A10-3-2-6-1			MaxSubpackets	If a host sends more Subpackets than the value of the TPer's MaxSubpackets property in a Packet on the Control Session, the Packet SHALL be discarded by the TPer.	Note: This test is only for the Control Session. There is no test for Regular Sessions because the behavior definition in [1] is a “MAY” statement for this case.
A10-3-2-15-1			MaxAuthentications	The value of the TPer's MaxAuthentications property SHALL NOT be 1.	

3.1.10. A11. StartSession()/SyncSession()

Upon successful invocation of StartSession using an authority with an Operation column value of Password, the Tries column of the C_PIN credential associated with that authority SHALL be reset to 0..

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
A11. StartSession()/SyncSession()					
A11-1-1-1-1	SessionID			If PacketHeader.Session is not all 0s, but a number of an open session, the StartSession() invocation SHALL fail because of 'invalid methodID' or 'invalid InvokingID' error leading to NOT_AUTHORIZED error.	
A11-3-2-1-1	Method invocation	Required Parameters	'HostSessionID'	If 'HostSessionID' indicates any number that can be encoded by 4-byte-long uinteger (i.e. is less than or equal to 2^{32}-1), the method invocation SHALL NOT fail because of that.	
A11-3-2-1-3				If 'HostSessionID' parameter indicates a number that cannot be encoded with 4-byte-length uinteger (i.e. is greater than 2^{32}-1), the drive SHALL ignore the method invocation and no response for that method SHALL be prepared.	
A11-3-2-2-2			'SPID'	If 'SPID' parameter indicates an UID that is nonexistent within the SP table in the Admin SP, the method invocation SHALL fail with a SyncSession formatted as follows: - The HostSessionID parameter SHALL be set to the HostSessionID that was sent by the host in StartSession - The SPSessionID parameter SHALL be set to a value such that the combination of HostSessionID and SPSessionID does not match any open sessions for that ComID, and - The SyncSession status SHALL be INVALID_PARAMETER.	
A11-3-2-2-3				If 'SPID' parameter indicates an SP that is in Manufactured-Inactive state, the method invocation SHALL fail with a SyncSession formatted as follows: - The HostSessionID parameter SHALL be set to the HostSessionID that was sent by the host in StartSession - The SPSessionID parameter SHALL be set to a value such that the combination of HostSessionID and SPSessionID does not match any open sessions for that ComID, and - The SyncSession status SHALL be INVALID_PARAMETER.	
A11-3-2-3-3(2)			'Write'	If 'Write' parameter indicates a value of 1 (and all other factors satisfy all conditions to make the method invocation succeed), the method SHALL NOT fail because of that.	

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
A11-3-2-3-4				If 'Write' parameter indicates a value other than 0 or 1, the method invocation SHALL fail with a SyncSession formatted as follows: - The HostSessionID parameter SHALL be set to the HostSessionID that was sent by the host in StartSession - The SPSessionID parameter SHALL be set to a value such that the combination of HostSessionID and SPSessionID does not match any open sessions for that ComID, and - The SyncSession status SHALL be INVALID_PARAMETER.	
A11-3-4-1-5		Optional Parameters	'HostChallenge' parameter	If the authority that is requested in 'HostSigningAuthority' requires Password authentication, if 'HostChallenge' parameter indicates the correct credential, and if the value of the TryLimit column in the corresponding object is set to 0 or the value of the Tries column is less than the value of the TryLimit column, the method invocation SHALL NOT fail because of that.	
A11-3-4-1-6				If the authority that is requested in 'HostSigningAuthority' requires Password authentication, if 'HostChallenge' parameter indicates the correct credential, and if the value of the TryLimit column in the corresponding object is set to non-0 and the value of the Tries column is equal to the value of the TryLimit column, the method invocation SHALL fail with a SyncSession formatted as follows: - The HostSessionID parameter SHALL be set to the HostSessionID that was sent by the host in StartSession - The SPSessionID parameter SHALL be set to a value such that the combination of HostSessionID and SPSessionID does not match any open sessions for that ComID, and - The SyncSession status SHALL be NOT_AUTHORIZED or AUTHORITY_LOCKED_OUT.	
A11-3-4-1-7				If 'HostChallenge' parameter indicates an incorrect credential, the method invocation SHALL fail with a SyncSession formatted as follows: - The HostSessionID parameter SHALL be set to the HostSessionID that was sent by the host in StartSession - The SPSessionID parameter SHALL be set to a value such that the combination of HostSessionID and SPSessionID does not match any open sessions for that ComID, and - The SyncSession status SHALL be NOT_AUTHORIZED. If the value of the TryLimit column is non-zero, then the value of the Tries column of corresponding object to the credential SHALL be incremented unless the value of the Tries column has reached the value of corresponding TryLimit column.	

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
A11-3-4-1-10				If the method invocation indicates session start-up with the Anybody authority explicitly in the HostSigningAuthority parameter, and if 'HostChallenge' parameter is encoded with some credential with proper encoding, i.e. byte token less than or equal to 32 byte length, the method invocation SHALL NOT fail because of that.	
A11-3-4-1-11				If the method invocation indicates session start-up with any authority that requires password authentication, and if 'HostChallenge' parameter is omitted, the method invocation SHALL fail with a SyncSession formatted as follows: - The HostSessionID parameter SHALL be set to the HostSessionID that was sent by the host in StartSession - The SPSessionID parameter SHALL be set to a value such that the combination of HostSessionID and SPSessionID does not match any open sessions for that ComID, and - The SyncSession status SHALL be INVALID_PARAMETER.	
A11-3-4-2-6			HostSigningAuthority' parameter	If 'HostSigningAuthority' parameter indicates an UID that is nonexistent within the Authority table in the requested SP, the method invocation SHALL fail with a SyncSession formatted as follows: - The HostSessionID parameter SHALL be set to the HostSessionID that was sent by the host in StartSession - The SPSessionID parameter SHALL be set to a value such that the combination of HostSessionID and SPSessionID does not match any open sessions for that ComID, and - The SyncSession status SHALL be INVALID_PARAMETER.	
A11-3-4-2-6(2)				If 'HostSigningAuthority' parameter indicates an authority's UID that is disabled, the method invocation SHALL fail with a SyncSession formatted as follows: - The HostSessionID parameter SHALL be set to the HostSessionID that was sent by the host in StartSession - The SPSessionID parameter SHALL be set to a value such that the combination of HostSessionID and SPSessionID does not match any open sessions for that ComID, and - The SyncSession status SHALL be INVALID_PARAMETER.	
A11-3-4-2-6(3)				If 'HostSigningAuthority' parameter indicates a class authority's UID, the method invocation SHALL fail with a SyncSession formatted as follows: - The HostSessionID parameter SHALL be set to the HostSessionID that was sent by the host in StartSession - The SPSessionID parameter SHALL be set to a value such that the combination of HostSessionID and SPSessionID does not match any open sessions for that ComID, and	

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
				- The SyncSession status SHALL be INVALID_PARAMETER.	
A11-3-4-2-9				If the 'HostSigningAuthority' parameter and 'HostChallenge' parameter are omitted, the method invocation SHALL NOT fail because of that and Anybody authority SHALL be authenticated.	
A11-3-5-6-1-1		session startup attempt during an open session existent		If this method invocation occurs when the number of open regular session(s) has already reached to the value of MaxSessions property, the method invocation SHALL fail with a SyncSession formatted as follows: - The HostSessionID parameter SHALL be set to the HostSessionID that was sent by the host in StartSession - The SPSessionID parameter SHALL be set to a value such that the combination of HostSessionID and SPSessionID does not match any open sessions for that ComID, and - The SyncSession status SHALL be NO_SESSIONS_AVAILABLE or SP_BUSY.	

3.1.11. A12. Get() - Basic Grammar -

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
A12. Get() - Basic Grammar -					
A12-0-1-1-1	Method invocation -General-	Required Parameter	'Table' component	If the component named 'Table', with component number of 0, is encoded, the method invocation SHALL fail with INVALID_PARAMETER.	
A12-0-1-1-2			Same component encoding	If the same component is encoded twice or more in a method invocation, the method invocation SHALL fail with INVALID_PARAMETER.	
A12-1-1-4-5(2)	Method invocation -Byte table-	Required Parameter	'startRow' component	If 'startRow' component indicates an address that is not within the limit of the table, the method invocation SHALL fail with INVALID_PARAMETER. (Note: Each limit of Byte tables can be identified by getting the corresponding value of Rows column in the Table table.)	
A12-1-1-4-10				If 'startRow' component's value is omitted, the method invocation SHALL behave the same as when 'startRow' component indicates 0.	
A12-1-1-5-6			'endRow' component	If 'endRow' component indicates an address that is not within the limit of the table, the method invocation SHALL fail with INVALID_PARAMETER. (Note: Each limit of Byte tables can be identified by getting the corresponding value of Rows column in the Table table.)	
A12-1-1-5-10				If 'endRow' component's value is omitted, the method invocation SHALL behave the same as when 'endRow' component indicates the end address of the table.	
A12-1-1-5-11			'startRow' - 'endRow' combination	If 'endRow' component is encoded prior to 'startRow' component, and 'startRow' is encoded, the method invocation SHALL fail with INVALID_PARAMETER.	
A12-1-1-5-12				If both 'startRow' and 'endRow' are encoded, and 'startRow' component indicates a larger value than the value 'endRow' component indicates, the method invocation SHALL fail with INVALID_PARAMETER.	
A12-1-1-6-1			'startColumn' component	If 'startColumn' component is encoded, the method invocation SHALL fail with INVALID_PARAMETER.	
A12-1-1-7-1			'endColumn' component	If 'endColumn' component is encoded, the method invocation SHALL fail with INVALID_PARAMETER.	
A12-3-1-4-1	Method invocation -An object-	Required Parameter	'startRow' component	If 'startRow' component is encoded, the method invocation SHALL fail with INVALID_PARAMETER.	
A12-3-1-5-1			'endRow' component	If 'endRow' component is encoded, the method SHALL fail with INVALID_PARAMETER.	

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
A12-3-1-6-6			'startColumn' component	If 'startColumn' component indicates any ColumnNumber that is larger than the highest column number in that table/object, the method invocation SHALL fail with INVALID_PARAMETER.	
A12-3-1-6-10				If 'startColumn' component is omitted, the method invocation SHALL behave the same as when 'startColumn' component indicates 0.	
A12-3-1-7-6			'endColumn' component	If 'endColumn' component indicates a ColumnNumber that is larger than the highest column number in that table/object, the method invocation SHALL fail with INVALID_PARAMETER.	
A12-3-1-7-9				If 'endColumn' component is omitted, the method invocation SHALL behave the same as when 'endColumn' component indicates the last column of the object.	
A12-3-1-7-10			'startColumn' - 'endColumn' combination	If 'endColumn' component is encoded prior to 'startColumn' component, and 'startColumn' is encoded, the method invocation SHALL fail with INVALID_PARAMETER.	
A12-3-1-7-10(2)				If both 'startColumn' and 'endColumn' are encoded and 'startColumn' component indicates a larger value than the value the 'endColumn' component indicates, the method invocation SHALL fail with INVALID_PARAMETER.	

3.1.12. A13. Set() - Basic Grammar -

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
A13. Set() -Basic Grammar-					
A13-2-1-2-5	Method invocation -Byte table-	Optional Parameter	'Where' parameter	If 'Where' parameter's value indicates an address within the limit of the Byte table, the method invocation SHALL NOT fail because of that.	
A13-2-1-2-6				If 'Where' parameter's value indicates an address that is not within the limit of the Byte table, the method invocation SHALL fail with INVALID_PARAMETER.	
A13-2-1-2-9				If 'Where' parameter is omitted, the method invocation SHALL behave the same as if the 'Where' parameter indicates 0.	
A13-2-1-3-5			'Values' parameter	If 'Values' parameter's value is encoded with byte token type, and contains a length that is fit into the limit of the table, the method SHALL NOT fail because of that.	
A13-2-1-3-6				If 'Values' parameter contains any data not to be addressed within the limit of the Byte table, the method invocation SHALL fail with INVALID_PARAMETER. (Note: Each limit of Byte tables can be identified by getting the corresponding value of Rows column in the Table table.)	
A13-2-1-3-9				If 'Values' parameter's value's length is equal to zero, the method invocation SHALL NOT fail, and SHALL NOT modify any cell and return SUCCESS status.	
A13-4-1-2-1	Method invocation -An object-	Optional Parameter	'Where' parameter	If 'Where' parameter is encoded, the method invocation SHALL fail with INVALID_PARAMETER.	
A13-4-1-4-14				If ColumnName-Value pairs are not encoded in ascending order with regard to the column number, the method invocation SHALL NOT fail because of that.	
A13-4-1-4-15				If ColumnName-Value pairs which indicate the same cell's modification are encoded in a method invocation, the method invocation SHALL fail with INVALID_PARAMETER.	

3.1.13. A14. Next() - Basic Grammar -

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
A14. Next() - Basic Grammar -					
A14-1-3-2-5(2)	Method invocation	Optional Parameter	'Where' parameter	If 'Where' parameter indicates an existing UID in the table, the method invocation SHALL NOT fail because of that.	
A14-1-3-2-8				If 'Where' parameter indicates a nonexistent UID in the table, the method invocation SHALL fail with INVALID_PARAMETER.	
A14-1-3-2-11				If 'Where' parameter is omitted, the method invocation SHALL return a list of zero or more UIDs, starting with the UID of the first row in the current ordering of the table. (Note that if 'Count' parameter indicates 0, no UID SHALL be sent by the TPer.)	
A14-1-3-3-6			'Count' parameter	If 'Count' parameter indicates a larger uinteger than the number of all the UIDs that follow the UID the 'Where' indicates in the order of UIDs for Next()'s responses defined in the table, the method invocation SHALL behave the same as when 'Count' component indicates the number of all the following UIDs	
A14-1-3-3-6(2)				If 'Count' parameter indicates 0, the method invocation SHALL NOT fail because of that, and SHALL NOT return any UID.	
A14-1-3-3-10				If 'Count' parameter is omitted, the method invocation SHALL behave the same as when 'Count' component indicates the number of all the UIDs that follow the UID the 'Where' indicates in the order of UIDs for Next()'s responses defined in the table.	

3.1.14. A15. GetACL() - Basic Grammar -

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
A15. GetACL() - Basic Grammar -					
A15-1-1-0-1	Method invocation	Condition		If the InvokingID, after the Call token ('F8'), is not the UID of AccessControl table, i.e. not 0x00000007 00000000, the method invocation SHALL fail with NOT_AUTHORIZED.	
A15-1-2-1-1(2)		Required Parameter	InvokingID	If 'InvokingID' parameter indicates an existing UID in the SP, and the UID is encoded by an 8-byte-long byte atom, a short or medium or long atom, the method invocation SHALL NOT fail because of that.	
A15-1-2-2-1(2)			MethodID	If 'MethodID' parameter indicates an existing MethodID in the SP, and the MethodID is encoded by an 8-byte-long byte atom, a short or medium or long atom, the method invocation SHALL NOT fail because of that.	
A15-1-2-3-1			'InvokingID' - 'MethodID' combination	If the combination of 'InvokingID' parameter's value and 'MethodID' parameter's value is nonexistence in the AccessControl table in the SP, the method invocation SHALL fail with NOT_AUTHORIZED.	

3.1.15. A19. RevertSP() - Basic Grammar -

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
A19. RevertSP() - Basic Grammar -					
A19-1-3-1-10	Method invocation	Optional Parameter	KeepGlobalRangeKey parameter	If 'KeepGlobalRangeKey' parameter is omitted, the method invocation SHALL behave the same as when 'KeepGlobalRangeKey' parameter indicates 0 uinteger.	

3.2. Section C: Table Content

3.2.1. C1. Level 0 Discovery Contents

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
C1. Level 0 Discovery Contents					
C1	Original Factory State	Verify of the transferred data		The contents of the Level 0 Discovery response data SHALL have the proper values specified by the Opal SSC[3].	

3.2.2. C2. Properties() Contents

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
C2. Properties() Contents					
C2	Original Factory State	Properties parameter		The contents of the response of a Properties() contains all the properties in the Properties parameter that are required by the Opal SSC[3], and their values meet the requirements in the SSC.	
C2(1)		HostProperties parameter		The contents of the response of a Properties() contains all the properties in the HostProperties parameter that are required by the Opal SSC[3], and their values meet the requirements in the SSC.	

3.2.3. C3. Get() Table Contents

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
C3. Get() Table Contents					
C3-1	AdminSP in its Original Factory State	Verify the table contents	Table	The contents of the data which could be retrieved by Get() invokings to all the columns and objects which is required by the Opal SSC[3] on this table SHALL have the values as specified the SSC.	
C3-2			SPInfo	The contents of the data which could be retrieved by Get() invokings to all the columns and objects which is required by the Opal SSC[3] on this table SHALL have the values as specified the SSC.	
C3-3			SPTemplates	The contents of the data which could be retrieved by Get() invokings to all the columns and objects which is required by the Opal SSC[3] on this table SHALL have the values as specified the SSC.	
C3-4			MethodID	The contents of the data which could be retrieved by Get() invokings to all the columns and objects which is required by the Opal SSC[3] on this table SHALL have the values as specified the SSC.	
C3-6			ACE	The contents of the data which could be retrieved by Get() invokings to all the columns and objects which is required by the Opal SSC[3] on this table SHALL have the values as specified the SSC.	
C3-7			Authority	The contents of the data which could be retrieved by Get() invokings to all the columns and objects which is required by the Opal SSC[3] on this table SHALL have the values as specified the SSC.	
C3-8			C_PIN	The contents of the data which could be retrieved by Get() invokings to all the columns and objects which is required by the Opal SSC[3]on this table SHALL have the values as specified the SSC.	
C3-9			TperInfo	The contents of the data which could be retrieved by Get() invokings to all the columns and objects which is required by the Opal SSC[3] on this table SHALL have the values as specified the SSC.	
C3-10			Template	The contents of the data which could be retrieved by Get() invokings to all the columns and objects which is required by the Opal SSC[3] on this table SHALL have the values as specified the SSC.	
C3-11			SP	The contents of the data which could be retrieved by Get() invokings to all the columns and objects which is required by the Opal SSC[3] on this table SHALL have the values as specified the SSC.	
C3-12	Locking SP just after being activated	Verify the table contents	Table	The contents of the data which could be retrieved by Get() invokings to all the columns and objects which is required by the Opal SSC[3] on this table SHALL have the values as specified the SSC.	

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
C3-13			SPInfo	The contents of the data which could be retrieved by Get() invokings to all the columns and objects which is required by the Opal SSC[3] on this table SHALL have the values as specified the SSC.	
C3-14			SPTemplates	The contents of the data which could be retrieved by Get() invokings to all the columns and objects which is required by the Opal SSC[3] on this table SHALL have the values as specified the SSC.	
C3-16			MethodID	The contents of the data which could be retrieved by Get() invokings to all the columns and objects which is required by the Opal SSC[3] on this table SHALL have the values as specified the SSC.	
C3-18			ACE	The contents of the data which could be retrieved by Get() invokings to all the columns and objects which is required by the Opal SSC[3] on this table SHALL have the values as specified the SSC.	
C3-19			Authority	The contents of the data which could be retrieved by Get() invokings to all the columns and objects which is required by the Opal SSC[3] on this table SHALL have the values as specified the SSC.	
C3-20			C_PIN	The contents of the data which could be retrieved by Get() invokings to all the columns and objects which is required by the Opal SSC[3] on this table SHALL have the values as specified the SSC.	
C3-21			LockingInfo	The contents of the data which could be retrieved by Get() invokings to all the columns and objects which is required by the Opal SSC[3] on this table SHALL have the values as specified the SSC.	
C3-22			Locking	The contents of the data which could be retrieved by Get() invokings to all the columns and objects which is required by the Opal SSC[3] on this table SHALL have the values as specified the SSC.	
C3-23			MBRControl	The contents of the data which could be retrieved by Get() invokings to all the columns and objects which is required by the Opal SSC[3] on this table SHALL have the values as specified the SSC.	
C3-24			MBR	The contents of the data which could be retrieved by Get() invokings to all the columns and objects which is required by the Opal SSC[3] on this table SHALL have the values as specified the SSC.	
C3-25			K_AES_128/K_AES_256	The contents of the data which could be retrieved by Get() invokings to all the columns and objects which is required by the Opal SSC[3] on this table SHALL have the values as specified the SSC.	
C3-26			DataStore	The contents of the data which could be retrieved by Get() invokings to all the columns and objects which is required by the Opal SSC[3] on this table SHALL have the values as specified the SSC.	

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
C3-27			RestrictedCommands	The contents of the data which could be retrieved by Get() invokings to all the columns and objects which is required by the Opal SSC[3]on this table SHALL have the values as specified the SSC.	This test is only for the TPers which support the RestrictedCommands table.

3.2.4. C4. Next() Contents

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
C4. Next() Contents					
C4-1	AdminSP in its Original Factory State	Verify the contents Next() retrieves	Table	All the UIDs of the data retrieved by a Next() invoking to this table without encoding of Where parameter and Count parameter SHALL contain all the UIDs the Opal SSC[3] requires.	
C4-3			SPTemplates	All the UIDs of the data retrieved by a Next() invoking to this table without encoding of Where parameter and Count parameter SHALL contain all the UIDs the Opal SSC[3] requires.	
C4-4			MethodID	All the UIDs of the data retrieved by a Next() invoking to this table without encoding of Where parameter and Count parameter SHALL contain all the UIDs the Opal SSC[3] requires.	
C4-6			ACE	All the UIDs of the data retrieved by a Next() invoking to this table without encoding of Where parameter and Count parameter SHALL contain all the UIDs the Opal SSC[3] requires.	
C4-7			Authority	All the UIDs of the data retrieved by a Next() invoking to this table without encoding of Where parameter and Count parameter SHALL contain all the UIDs the Opal SSC[3] requires.	
C4-8			C_PIN	All the UIDs of the data retrieved by a Next() invoking to this table without encoding of Where parameter and Count parameter SHALL contain all the UIDs the Opal SSC[3] requires.	
C4-10			Template	All the UIDs of the data retrieved by a Next() invoking to this table without encoding of Where parameter and Count parameter SHALL contain all the UIDs the Opal SSC[3] requires.	
C4-11			SP	All the UIDs of the data retrieved by a Next() invoking to this table without encoding of Where parameter and Count parameter SHALL contain all the UIDs the Opal SSC[3] requires.	
C4-12	Locking SP just after being activated	Verify the contents Next() retrieves	Table	All the UIDs of the data retrieved by a Next() invoking to this table without encoding of Where parameter and Count parameter SHALL contain all the UIDs the Opal SSC[3] requires.	
C4-14			SPTemplates	All the UIDs of the data retrieved by a Next() invoking to this table without encoding of Where parameter and Count parameter SHALL contain all the UIDs the Opal SSC[3] requires.	
C4-16			MethodID	All the UIDs of the data retrieved by a Next() invoking to this table without encoding of Where parameter and Count parameter SHALL contain all the UIDs the Opal SSC[3] requires.	

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
C4-18			ACE	All the UIDs of the data retrieved by a Next() invoking to this table without encoding of Where parameter and Count parameter SHALL contain all the UIDs the Opal SSC[3] requires.	
C4-19			Authority	All the UIDs of the data retrieved by a Next() invoking to this table without encoding of Where parameter and Count parameter SHALL contain all the UIDs the Opal SSC[3] requires.	
C4-20			C_PIN	All the UIDs of the data retrieved by a Next() invoking to this table without encoding of Where parameter and Count parameter SHALL contain all the UIDs the Opal SSC[3] requires.	
C4-22			Locking	All the UIDs of the data retrieved by a Next() invoking to this table without encoding of Where parameter and Count parameter SHALL contain all the UIDs the Opal SSC[3] requires.	
C4-23			RestrictedCommands	All the UIDs of the data retrieved by a Next() invoking to this table without encoding of Where parameter and Count parameter SHALL contain all the UIDs the Opal SSC[3] requires.	This test is only for the TPers which support the RestrictedCommands table.

3.2.5. C5. GetACL() Contents

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
C5. GetACL() Contents					
C5-1	AdminSP in its Original Factory State	Verify the AccessControl table's ACL column values	Table	The values of the ACLs retrieved by GetACL() invokings with all the combinations of the InvokingID parameter and MethodID parameter which are required by the SSC SHALL have the values as specified by the SSC.	
C5-2			SPInfo	The values of the ACLs retrieved by GetACL() invokings with all the combinations of the InvokingID parameter and MethodID parameter which are required by the SSC SHALL have the values as specified by the SSC.	
C5-3			SPTemplates	The values of the ACLs retrieved by GetACL() invokings with all the combinations of the InvokingID parameter and MethodID parameter which are required by the SSC SHALL have the values as specified by the SSC.	
C5-4			MethodID	The values of the ACLs retrieved by GetACL() invokings with all the combinations of the InvokingID parameter and MethodID parameter which are required by the SSC SHALL have the values as specified by the SSC.	
C5-5			ACE	The values of the ACLs retrieved by GetACL() invokings with all the combinations of the InvokingID parameter and MethodID parameter which are required by the SSC SHALL have the values as specified by the SSC.	
C5-6			Authority	The values of the ACLs retrieved by GetACL() invokings with all the combinations of the InvokingID parameter and MethodID parameter which are requiredby the SSC SHALL have the values as specified by the SSC.	
C5-7			C_PIN	The values of the ACLs retrieved by GetACL() invokings with all the combinations of the InvokingID parameter and MethodID parameter which are required by the SSC SHALL have the values as specified by the SSC.	
C5-8			TperInfo	The values of the ACLs retrieved by GetACL() invokings with all the combinations of the InvokingID parameter and MethodID parameter which are required by the SSC SHALL have the values as specified by the SSC.	
C5-9			Template	The values of the ACLs retrieved by GetACL() invokings with all the combinations of the InvokingID parameter and MethodID parameter which are required by the SSC SHALL have the values as specified by the SSC.	
C5-10			SP	The values of the ACLs retrieved by GetACL() invokings with all the combinations of the InvokingID parameter and MethodID parameter which are required by the SSC SHALL have the values as specified by the SSC.	
C5-11	Locking SP just after being activated	Verify the AccessControl table's ACL column values	Table	The values of the ACLs retrieved by GetACL() invokings with all the combinations of the InvokingID parameter and MethodID parameter which are required by the SSC SHALL have the values as specified by the SSC.	

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
C5-12			SPInfo	The values of the ACLs retrieved by GetACL() invokings with all the combinations of the InvokingID parameter and MethodID parameter which are required by the SSC SHALL have the values as specified by the SSC.	
C5-13			SPTemplates	The values of the ACLs retrieved by GetACL() invokings with all the combinations of the InvokingID parameter and MethodID parameter which are required by the SSC SHALL have the values as specified by the SSC.	
C5-15			MethodID	The values of the ACLs retrieved by GetACL() invokings with all the combinations of the InvokingID parameter and MethodID parameter which are required by the SSC SHALL have the values as specified by the SSC.	
C5-16			ACE	The values of the ACLs retrieved by GetACL() invokings with all the combinations of the InvokingID parameter and MethodID parameter which are required by the SSC SHALL have the values as specified by the SSC.	
C5-17			Authority	The values of the ACLs retrieved by GetACL() invokings with all the combinations of the InvokingID parameter and MethodID parameter which are required by the SSC SHALL have the values as specified by the SSC.	
C5-18			C_PIN	The values of the ACLs retrieved by GetACL() invokings with all the combinations of the InvokingID parameter and MethodID parameter which are required by the SSC SHALL have the values as specified by the SSC.	
C5-19			LockingInfo	The values of the ACLs retrieved by GetACL() invokings with all the combinations of the InvokingID parameter and MethodID parameter which are required by the SSC SHALL have the values as specified by the SSC.	
C5-20			Locking	The values of the ACLs retrieved by GetACL() invokings with all the combinations of the InvokingID parameter and MethodID parameter which are required by the SSC SHALL have the values as specified by the SSC.	
C5-21			MBRControl	The values of the ACLs retrieved by GetACL() invokings with all the combinations of the InvokingID parameter and MethodID parameter which are required by the SSC SHALL have the values as specified by the SSC.	
C5-22			MBR	The values of the ACLs retrieved by GetACL() invokings with all the combinations of the InvokingID parameter and MethodID parameter which are required by the SSC SHALL have the values as specified by the SSC.	
C5-23			K_AES_128/K_AES_256	The values of the ACLs retrieved by GetACL() invokings with all the combinations of the InvokingID parameter and MethodID parameter which are required by the SSC SHALL have the values as specified by the SSC.	
C5-24			DataStore	The values of the ACLs retrieved by GetACL() invokings with all the combinations of the InvokingID parameter and MethodID parameter which are required by the SSC SHALL have the values as specified by the SSC.	

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
C5-25			ThisSP	The values of the ACLs retrieved by GetACL() invokings with all the combinations of the InvokingID parameter and MethodID parameter which are required by the SSC SHALL have the values as specified by the SSC.	
C5-26			RestrictedCommands	The values of the ACLs retrieved by GetACL() invokings with all the combinations of the InvokingID parameter and MethodID parameter which are required by the SSC SHALL have the values as specified by the SSC.	

3.3. Section D: Grammer and Effect

3.3.1. D1. ACE.Set() - Grammar and Effect -

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
D1. ACE.Set() - Grammar and Effect -					
D1-1-1-1-9	Set() 'BooleanExpr' column	Grammar		If the Name-Value pairs' Value, where its Name indicates 'BooleanExpr' column, indicates any form that can be parsed as postfix boolean expression form, the method invocation SHALL no fail because of that.	
D1-1-1-1-10				If the Name-Value pairs' Value, where its Name indicates 'BooleanExpr' column, indicates any form that cannot be parsed as postfix boolean expression form, the method invocation SHALL lead to session abort.	
D1-1-1-1-11				If the Name-Value pairs' Value, where its Name indicates 'BooleanExpr' column, contains such number of Authority UIDs and boolean_ACE uintegers that is at most the maximum size of AC_element defined in each TPer, the method SHALL NOT fail because of that. (Note: The test shall attempt up to the minimum required by Opal[3], which is 9 entries.)	
D1-1-1-1-13				If the Name-Value pairs' Value, where its Name indicates 'BooleanExpr' column, contains such number of Authority UIDs and boolean_ACE uintegers that is more than the maximum size of AC_element defined in each TPer, the method SHALL fail with INVALID_PARAMETER. (Note: The test shall attempt up to the minimum required by Opal, which is 9 entries.)	
		Effects of successful method invocation	(Description) The successful invocation of this method SHALL cause, (and a non-successful invocation SHALL NOT cause), the effects as below:		
D1-1-2-1-1				The value of the column gets set into the value the Set() indicates. So, if a Get() is issued to the column, the expected value the Set() indicates SHALL be read.	
D1-1-2-1-2				The access control settings are modified as the Set() indicates.	
D1-1-3-1-1		Effects of this method invocation within a transaction		Set() for a 'BooleanExpr' column is invoked within a transaction. -> The column value and the corresponding access control setting become the new value that the Set() indicates immediately. EndTransaction token with zero status is sent for the top-level transaction. -> The column value and access control setting retains the set value.	

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
D1-1-3-1-2				Set() for a 'BooleanExpr' column is invoked within a transaction. -> The column value and the corresponding access control setting become the new value that the Set() indicates immediately. EndTransaction token with non-zero status is sent for the top-level transaction. -> The column value and access control setting change back to the value before the transaction.	

3.3.2. D2. Authority.Set() - Grammar and Effect -

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
D2. Authority.Set() - Grammar and Effect -					
	Set() 'Enabled' column	Effects of successful method invocation	(Description) The successful invocation of this method SHALL cause, (and a non-successful invocation SHALL NOT cause), the effects as below:		
D2-1-2-1-1			Common	The value of the column gets set into the value the Set() indicates. So, if a Get() is issued to the column, the expected value the Set() indicates SHALL be read.	
D2-1-2-2-1			If the value gets set to 'true'	Authenticating attempts to the SP that contains this authority with no grammatical violation, the correct credential, and indication of this authority SHALL succeed.	
D2-1-2-3-1			If the value gets set to 'false'	Authenticating attempts to the SP that contains this authority with no grammatical violation, the correct credential, and indication of this authority SHALL fail.	
D2-1-2-3-2				The previous successful authentication result with this authority within the same session SHALL NOT be affected by this, still valid.	
D2-1-3-1-1		Effects of this method invocation within a transaction		Set() for a 'Enabled' column is invoked within a transaction. -> The authority becomes enabled/disabled (depending on the set value) immediately. EndTransaction token with zero status is sent for the top-level transaction. -> The authority's enabled column and enabled status retain the set value.	
D2-1-3-1-2				Set() for a 'Enabled' column is invoked within a transaction. -> The authority becomes enabled/disabled (depending on the set value) immediately. EndTransaction token with non-zero status is sent for the top-level transaction. -> The authority's enabled column and enabled status change back to the value before the transaction.	

3.3.3. D3. C_PIN.Set() - Grammar and Effect -

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
D3. C_PIN.Set() - Grammar and Effect -					
	Set() PIN column	Effects of successful method invocation	(Description) The successful invocation of this method SHALL cause, (and a non-successful invocation SHALL NOT cause), the effects as below:		
D3-1-2-1-2				The correct credential for authentication changes the value as the Set() indicates. (When Null, 'A0', is indicated by the Set(), the correct credential for authentication changes Null.) (When 32-byte data is indicated by the Set(), the correct credential for authentication changes as indicated.)	
D3-1-3-1-1		Effects of this method invocation within a transaction		Set() for a 'PIN' column is invoked within a transaction. -> The PIN credential becomes the new value the Set() indicates immediately. EndTransaction token with zero status is sent for the top-level transaction. -> The PIN credential retains the set value.	
D3-1-3-1-2				Set() for a 'PIN' column is invoked within a transaction. -> The PIN credential becomes the new value the Set() indicates immediately. EndTransaction token with non-zero status is sent for the top-level transaction. -> The PIN credential changes back to the value before the transaction.	

3.3.4. D4. Locking.Set() - Grammar and Effect -

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
D4. Locking.Set() - Grammar and Effect -					
D4-1-2-1-2	Set() 'RangeStart' column and 'RangeLength' column	Condition		If the combination of 'RangeStart' and 'RangeLength' indicated by the Set() contains any LBA that overlaps with any other ranges' LBA except GlobalRange's, the method SHALL fail with INVALID_PARAMETER.	
		Effects of successful method invocation	(Description) The successful invocation of this method SHALL cause, (and a non-successful invocation SHALL NOT cause), the effects as below:		
D4-1-3-1-1			Common	The value of the column gets set into the value the Set() indicates. So, if a Get() is issued to the column, the Get() retrieves the value the Set() indicated.	
D4-1-3-2-1			If 'RangeStart' gets changed and 'RangeLength' is already non-0	The beginning and ending of LBAs in that range changes as intended by Set().	
D4-1-3-3-1			If 'RangeStart' gets changed and 'RangeLength' is already 0	No LBAs are covered by that range.	
D4-1-3-4-1			If 'RangeLength' gets set to non-0	The end of LBAs in that range changes as intended by Set().	
D4-1-3-5-1			If 'RangeLength' gets set to 0	No LBAs are covered by that range.	
D4-1-4-1-1		Effects of this method invocation within a transaction		Set() for a 'RangeStart' column or a 'RangeLength' column, or for both, is invoked within a transaction. -> The column value(s) become the new value that the Set() indicates immediately, but the corresponding Locking Range's start LBA and length are not affected. EndTransaction token with zero status is sent for the top-level transaction. -> The column value(s) retains the set values, and the start LBA and the length change as the set values indicate.	
D4-1-4-1-2				Set() for a 'RangeStart' column or a 'RangeLength' column, or for both, is invoked within a transaction. -> The column value(s) become the new value that the Set() indicates immediately, but the corresponding Locking Range's start LBA and length are not affected. EndTransaction token with non-zero status is sent for the top-level transaction. -> The column value(s) and change(s) of the start LBA and the length back to the value before the transaction started.	
	Set() 'ReadLockEnabled' column and Set() 'ReadLocked' column	Effects of successful method invocation	(Description) The successful invocation of this method SHALL cause, (and a non-successful invocation SHALL NOT cause), the effects as below:		

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
D4-2-2-1-1			Common	The value(s) of these columns get set into the value the Set() indicates. So, if a Get() is issued to the column, the expected value the Set() indicates SHALL be read.	
D4-2-2-2-1			When 'ReadLockEnabled' is 'true' and 'ReadLocked' is 'true'	When the TPer's MBR shadowing is not active, READ commands addressing ONLY LBA(s) covered by this range SHALL be aborted at the ATA interface level with: i) ST = 0x51 and ER = 0x04 or ii) ST has bit 1 set, Sense Key = DATA PROTECT and ASC/ASCQ = ACCESS DENIED – NO ACCESS RIGHTS if the TPer supports Sense Data Reporting feature set and it is enabled.	
D4-2-2-2-1(2)				When the TPer's MBR shadowing is not active, READ commands addressing LBA(s) covered by this range and LBA(s) covered by other range(s) SHALL be aborted at the ATA interface level with: i) ST = 0x51 and ER = 0x04 or ii) ST has bit 1 set, Sense Key = DATA PROTECT and ASC/ASCQ = ACCESS DENIED – NO ACCESS RIGHTS if the TPer supports Sense Data Reporting feature set and it is enabled.	
D4-2-2-2-2				When the TPer's MBR shadowing is active, READ commands addressing such LBAs that every one is covered by this range but not by the MBR SHALL be successfully returned with all-0 payload.	
D4-2-2-2-3				'Locked' bit in Level 0 discovery response data is set to 1.	
D4-2-2-2-5				Power-on reset SHALL cause 'ReadLocked' value to change into 'true' and the same effects appear as the effects appear when both 'ReadLockEnabled' and 'ReadLocked' are set to 'true'.	
D4-2-2-3-1			When 'ReadLockEnabled' is 'true' and 'ReadLocked' is 'false'	When the TPer's MBR shadowing is not active, READ commands addressing ONLY LBA(s) covered by this range SHALL be successfully returned with the actual data for the LBA(s) if there is not any other error factor.	
D4-2-2-3-1(2)				When the TPer's MBR shadowing is not active, READ commands spanning multiple LBA ranges including this range SHALL be handled as follows: If the ‘RangeCrossing’ bit in Level 0 Discovery response is 0, and all crossed ranges are not Read-Locked, crossing multiple ranges SHALL NOT cause the READ command to fail. If the ‘RangeCrossing’ bit in Level 0 Discovery response is 1, crossing multiple ranges SHALL cause the READ command to be aborted at the ATA interface level with: i) ST = 0x51 and ER = 0x04 or ii) ST has bit 1 set, Sense Key = ILLEGAL REQUEST and ASC/ASCQ = INVALID FIELD IN CDB if the TPer supports Sense Data Reporting feature set and it is enabled.	

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
D4-2-2-3-2				When the TPer's MBR shadowing is active, READ commands addressing ONLY LBA(s) covered by this range, but not LBA(s) covered by the MBR, SHALL be successfully returned with the actual data for the LBA(s) if there is not any other error factor.	
D4-2-2-3-3				If this range is not write-locked nor all the other ranges are not read-locked nor write-locked, 'Locked' bit in Level 0 discovery response data becomes 0. Otherwise 1.	
D4-2-2-3-5				Power-on reset SHALL cause 'ReadLocked' value to change into 'true' and the same effects appear as the effects appear when both 'ReadLockEnabled' and 'ReadLocked' are set to 'true'.	
D4-2-2-4-1			When 'ReadLockEnabled' is 'false' and 'ReadLocked' is either 'true' or 'false'	When the TPer's MBR shadowing is not active, READ commands addressing ONLY LBA(s) covered by this range SHALL be successfully returned with the actual data for the LBA(s) if there is not any other error factor.	
D4-2-2-4-1(2)				When the TPer's MBR shadowing is not active, READ commands spanning multiple LBA ranges including this range SHALL be handled as follows: If the 'RangeCrossing' bit in Level 0 Discovery response is 0, and all crossed ranges are not Read-Locked, crossing multiple ranges SHALL NOT cause the READ command to fail. If the 'RangeCrossing' bit in Level 0 Discovery response is 1, crossing multiple ranges SHALL cause the READ command to be aborted at the ATA interface level with: i) ST = 0x51 and ER = 0x04 or ii) ST has bit 1 set, Sense Key = ILLEGAL REQUEST and ASC/ASCQ = INVALID FIELD IN CDB if the TPer supports Sense Data Reporting feature set and it is enabled.	
D4-2-2-4-2				When the TPer's MBR shadowing is active, READ commands addressing ONLY LBA(s) covered by this range, but not LBA(s) covered by the MBR, SHALL be successfully returned with the actual data for the LBA(s) if there is not any other error factor.	
D4-2-2-4-3				If this range is not write-locked nor all the other ranges are not read-locked nor write-locked, 'Locked' bit in Level 0 discovery response data becomes 0. Otherwise 1.	
D4-2-3-1-1		Effects of this method invocation within a transaction		Set() for a 'ReadLockEnabled' column or a 'ReadLocked' column, or for both, is invoked within a transaction. -> The column value(s) become the new value(s) that the Set() indicates immediately, but the corresponding Locking Range's read-locking state and the behavior after a power cycle, are not affected unless the transaction commits successfully. EndTransaction token with zero status is sent for the top-level transaction. -> The column value(s) retains the set values, and the read-locking state and the behavior after a power cycle,	

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
				change as the column values indicate.	
D4-2-3-1-2				Set() for a 'ReadLockEnabled' column or a 'ReadLocked' column, or for both, is invoked within a transaction. -> The column value(s) become the new value(s) that the Set() indicates immediately, but the corresponding Locking Range's read-locking state and the behavior after a power cycle, are not affected unless the transaction commits successfully. EndTransaction token with non-zero status is sent for the top-level transaction. -> The column value(s), the read-locking state and the behavior after a power cycle back to the states before the transaction started.	
	Set() 'WriteLockEnabled' column and Set() 'WriteLocked' column	Effects of successful method invocation	(Description) The successful invocation of this method SHALL cause, (and a non-successful invocation SHALL NOT cause), the effects as below:		
D4-3-2-1-1			Common	The value(s) of these columns are set into the value the Set() indicates. So, if a Get() is issued to the column, the Get() retrieves the value the Set() indicated.	
D4-3-2-2-1			When 'WriteLockEnabled' is 'true' and 'WriteLocked' is 'true'	When the TPer's MBR shadowing is not active, WRITE commands addressing ONLY LBA(s) covered by this range SHALL be aborted at the ATA interface level with: i) ST = 0x51 and ER = 0x04 or ii) ST has bit 1 set, Sense Key = DATA PROTECT and ASC/ASCQ = ACCESS DENIED – NO ACCESS RIGHTS if the TPer supports Sense Data Reporting feature set and it is enabled.	
D4-3-2-2-1(2)				When the TPer's MBR shadowing is not active, WRITE commands addressing LBA(s) covered by this range and LBA(s) covered by other range(s) SHALL be aborted at the ATA interface level with: i) ST = 0x51 and ER = 0x04 or ii) ST has bit 1 set, Sense Key = DATA PROTECT and ASC/ASCQ = ACCESS DENIED – NO ACCESS RIGHTS if the TPer supports Sense Data Reporting feature set and it is enabled.	
D4-3-2-2-2				When the TPer's MBR shadowing is active, WRITE commands addressing ONLY LBA(s) covered by this range but not by the MBR SHALL be aborted at the ATA interface level with: i) ST = 0x51 and ER = 0x04 or	

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
				ii) ST has bit 1 set, Sense Key = DATA PROTECT and ASC/ASCQ = ACCESS DENIED – NO ACCESS RIGHTS if the TPer supports Sense Data Reporting feature set and it is enabled.	
D4-3-2-2-3				'Locked' bit in Level 0 discovery response data is set to 1.	
D4-3-2-2-5				Power-on reset SHALL cause 'WriteLocked' value to change into 'true' and the same effects appear as the effects appear when both 'WriteLockEnabled' and 'WriteLocked' are set to 'true'.	
D4-3-2-3-1			When 'WriteLockEnabled' is 'true' and 'WriteLocked' is 'false'	When the TPer's MBR shadowing is not active, WRITE commands addressing ONLY LBA(s) covered by this range SHALL NOT fail if there is not any other error factor.	
D4-3-2-3-1(2)				When the TPer's MBR shadowing is not active, WRITE commands spanning multiple LBA ranges including this range SHALL be handled as follows: If the ‘RangeCrossing’ bit in Level 0 Discovery response is 0, and all crossed ranges are not Write-Locked, crossing multiple ranges SHALL NOT cause the WRITE command to fail. If the ‘RangeCrossing’ bit in Level 0 Discovery response is 1, crossing multiple ranges SHALL cause the WRITE command to be aborted at the ATA interface level with: i) ST = 0x51 and ER = 0x04 or ii) ST has bit 1 set, Sense Key = ILLEGAL REQUEST and ASC/ASCQ = INVALID FIELD IN CDB if the TPer supports Sense Data Reporting feature set and it is enabled.	
D4-3-2-3-2				When the TPer's MBR shadowing is active, WRITE commands addressing LBA(s) covered by this range, but not LBA(s) covered by the MBR, SHALLNOT fail if there is not any other error factor.	
D4-3-2-3-3				If this range is not read-locked nor all the other ranges are not read-locked nor write-locked, 'Locked' bit in Level 0 discovery response data becomes 0. Otherwise 1.	
D4-3-2-3-5				Power-on reset will cause 'WriteLocked' value to change into 'true' and the same effects appear as the effects appear when both 'WriteLockEnabled' and 'WriteLocked' get set to 'true'.	
D4-3-2-4-1			When 'WriteLockEnabled' is 'false' and 'WriteLocked' is either ‘true’ or 'false'	When the TPer's MBR shadowing is not active, WRITE commands addressing ONLY LBA(s) covered by this range SHALL NOT fail if there is not any other error factor.	

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
D4-3-2-4-1(2)				When the TPer's MBR shadowing is not active, WRITE commands spanning multiple LBA ranges including this range SHALL be handled as follows: If the ‘RangeCrossing’ bit in Level 0 Discovery response is 0, and all crossed ranges are not Write-Locked, crossing multiple ranges SHALL NOT cause the WRITE command to fail. If the ‘RangeCrossing’ bit in Level 0 Discovery response is 1, crossing multiple ranges SHALL cause the WRITE command to be aborted at the ATA interface level with: i) ST = 0x51 and ER = 0x04 or ii) ST has bit 1 set, Sense Key = ILLEGAL REQUEST and ASC/ASCQ = INVALID FIELD IN CDB if the TPer supports Sense Data Reporting feature set and it is enabled.	
D4-3-2-4-2				When the TPer's MBR shadowing is active, WRITE commands addressing ONLY LBA(s) covered by this range, but not LBA(s) covered by the MBR SHALL NOT fail if there is not any other error factor.	
D4-3-2-4-3				If this range is not read-locked nor all the other ranges are not read-locked nor write-locked, 'Locked' bit in Level 0 discovery response data becomes 0. Otherwise 1.	
D4-3-3-1-1		Effects of this method invocation upon a transaction		Set() for a 'WriteLockEnabled' column or a 'WriteLocked' column, or for both, is invoked within a transaction. -> The column value(s) become the new value(s) that the Set() indicates immediately, but the corresponding Locking Range's write-locking state and the behavior after a power cycle, are not affected unless the transaction commits successfully. EndTransaction token with zero status is sent for the top-level transaction. -> The column value(s) retains the set value and the write-locking state and the behavior after a power cycle, change as the column values indicate.	
D4-3-3-1-2				Set() for a 'WriteLockEnabled' column or a 'WriteLocked' column, or for both, is invoked within a transaction. -> The column value(s) become the new value(s) that the Set() indicates immediately, but the corresponding Locking Range's write-locking state and the behavior after a power cycle, are not affected unless the transaction commits successfully. EndTransaction token with non-zero status is sent for the top-level transaction. -> The column value(s), the write-locking state and the behavior after a power cycle back to the states before the transaction started.	

3.3.5. D5. MBRControl.Set() - Grammar and Effect -

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
D5. MBRControl.Set() - Grammar and Effect -					
	Set() 'Enable' column and Set() 'Done' column	Effects of successful method invocation	(Description) The successful invocation of this method SHALL cause, (and a non-successful invocation SHALL NOT cause), the effects as below:		
D5-1-2-1-1			Common	The value(s) of these columns get set into the value the Set() indicates. So, if a Get() is issued to the column, the Get() SHALL retrieve the value(s) the Set() indicated.	
D5-1-2-2-1			When both 'Enable' and 'Done' are set to 'true'	Power-on reset SHALL cause 'Done' value to change into 'false'.	
D5-1-2-2-2				READ commands SHALL be aborted or successfully returned with actual data (assuming no other error), depending on the locking state of the ranges that cover requested LBAs, and depending on whether the requested LBA(s) span more than one ranges if the "RangeCrossing" is set to 1: i) If the command addresses an LBA in a read-locked Locking Range, the command SHALL be aborted at the ATA interface level with: i-a) ST = 0x51 and ER = 0x04 or i-b) ST has bit 1 set, Sense Key = DATA PROTECT and ASC/ASCQ = ACCESS DENIED – NO ACCESS RIGHTS if the TPer supports Sense Data Reporting feature set and it is enabled. ii) If the command doesn't address any LBA in a read-locked Locking Range, but spans multiple Locking Ranges when 'RangeCrossing' bit in Level 0 Discovery response is 1, the command SHALL be aborted at the ATA interface level with: ii-a) ST = 0x51 and ER = 0x04 or ii-b) ST has bit 1 set, Sense Key = ILLEGAL REQUEST and ASC/ASCQ = INVALID FIELD IN CDB if the TPer supports Sense Data Reporting feature set and it is enabled. iii) Otherwise the command SHALL succeed with actual data (assuming no other error).	
D5-1-2-2-3				WRITE commands SHALL be aborted or succeed (assuming no other error), depending on the locking state of the ranges that cover requested LBAs, and depending on whether the requested LBA(s) span more than one ranges if the "RangeCrossing" is set to 1: i) If the command addresses an LBA in a write-locked Locking Range, the command SHALL be aborted at the ATA interface level with: i-a) ST = 0x51 and ER = 0x04 or i-b) ST has bit 1 set, Sense Key = DATA PROTECT and ASC/ASCQ = ACCESS DENIED – NO ACCESS RIGHTS if the TPer supports Sense Data Reporting feature set and it is enabled.	

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
				ii) If the command doesn't address any LBA in a write-locked Locking Range, but spans multiple Locking Ranges when 'RangeCrossing' bit in Level 0 Discovery response is 1, the command SHALL be aborted at the ATA interface level with: ii-a) ST = 0x51 and ER = 0x04 or ii-b) ST has bit 1 set, Sense Key = ILLEGAL REQUEST and ASC/ASCQ = INVALID FIELD IN CDB if the TPer supports Sense Data Reporting feature set and it is enabled. iii) Otherwise the command SHALL succeed (assuming no other error).	
D5-1-2-2-4				'MBRDone' bit in Level 0 discovery response data is set to 1.	
D5-1-2-2-4(2)				'MBREnabled' bit in Level 0 discovery response data is set to 1.	
D5-1-2-3-1			When 'Enable' is set to 'true' and 'Done' is set to 'false'	Power-on reset SHALL cause 'Done' value to change into 'false'.	
D5-1-2-3-2				READ commands addressing ONLY LBA(s) covered by the MBR, SHALL be successfully returned with the data stored in the MBR if there is not any other error factor.	
D5-1-2-3-2(2)				READ commands addressing LBA(s) covered by the MBR and LBA(s) not covered by the MBR, SHALL be aborted at the ATA interface level with: i) ST = 0x51 and ER = 0x04 or ii) ST has bit 1 set, Sense Key = DATA PROTECT and ASC/ASCQ = ACCESS DENIED – NO ACCESS RIGHTS if the TPer supports Sense Data Reporting feature set and it is enabled.	
D5-1-2-3-3				WRITE commands addressing ONLY LBA(s) covered by the MBR SHALL be aborted at the ATA interface level with: i) ST = 0x51 and ER = 0x04 or ii) ST has bit 1 set, Sense Key = DATA PROTECT and ASC/ASCQ = ACCESS DENIED – NO ACCESS RIGHTS if the TPer supports Sense Data Reporting feature set and it is enabled.	
D5-1-2-3-3(2)				WRITE commands addressing LBA(s) covered by the MBR and LBA(s) not covered by the MBR, SHALL be aborted at the ATA interface level with: i) ST = 0x51 and ER = 0x04 or ii) ST has bit 1 set, Sense Key = DATA PROTECT and ASC/ASCQ = ACCESS DENIED – NO ACCESS RIGHTS if the TPer supports Sense Data Reporting feature set and it is enabled.	
D5-1-2-3-6				'MBRDone' bit in Level 0 discovery response data response data is set to 0.	
D5-1-2-3-6(2)				'MBREnabled' bit in Level 0 discovery response data is set to 1.	

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
D5-1-2-4-1			When 'Enable' is set to 'false' and 'Done' is set to either 'true' or 'false'	READ commands SHALL be aborted or successfully returned with actual data (assuming no other error), depending on the locking state of the ranges that cover requested LBAs, and depending on whether the requested LBA(s) span more than one ranges if the "RangeCrossing" is set to 1: i) If the command addresses an LBA in a read-locked Locking Range, the command SHALL be aborted at the ATA interface level with: i-a) ST = 0x51 and ER = 0x04 or i-b) ST has bit 1 set, Sense Key = DATA PROTECT and ASC/ASCQ = ACCESS DENIED – NO ACCESS RIGHTS if the TPer supports Sense Data Reporting feature set and it is enabled. ii) If the command doesn’t address any LBA in a read-locked Locking Range, but spans multiple Locking Ranges when ‘RangeCrossing’ bit in Level 0 Discovery response is 1, the command SHALL be aborted at the ATA interface level with: ii-a) ST = 0x51 and ER = 0x04 or ii-b) ST has bit 1 set, Sense Key = ILLEGAL REQUEST and ASC/ASCQ = INVALID FIELD IN CDB if the TPer supports Sense Data Reporting feature set and it is enabled. iii) Otherwise the command SHALL succeed with actual data (assuming no other error).	
D5-1-2-4-2				WRITE commands SHALL be aborted or succeed (assuming no other error), depending on the locking state of the ranges that cover requested LBAs, and depending on whether the requested LBA(s) span more than one ranges if the "RangeCrossing" is set to 1: i) If the command addresses an LBA in a write-locked Locking Range, the command SHALL be aborted at the ATA interface level with: i-a) ST = 0x51 and ER = 0x04 or i-b) ST has bit 1 set, Sense Key = DATA PROTECT and ASC/ASCQ = ACCESS DENIED – NO ACCESS RIGHTS if the TPer supports Sense Data Reporting feature set and it is enabled. ii) If the command doesn’t address any LBA in a write-locked Locking Range, but spans multiple Locking Ranges when ‘RangeCrossing’ bit in Level 0 Discovery response is 1, the command SHALL be aborted at the ATA interface level with: ii-a) ST = 0x51 and ER = 0x04 or ii-b) ST has bit 1 set, Sense Key = ILLEGAL REQUEST and ASC/ASCQ = INVALID FIELD IN CDB if the TPer supports Sense Data Reporting feature set and it is enabled. iii) Otherwise the command SHALL succeed (assuming no other error).	
D5-1-2-4-3				'MBRDone' bit in Level 0 discovery response data is set to 0.	
D5-1-2-4-3(2)				'MBREnabled' bit in Level 0 discovery response data is set to 0.	

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
D5-1-3-1-1		Effects of this method invocation within a transaction		Set() for a 'Enable' column or a 'Done' column, or for both, is invoked within a transaction. -> The column value(s) become the new value(s) that the Set() indicates immediately, but the mode of the TPer, i.e. whether in MBR mode or not, and the behavior after a power cycle, are not affected unless the transaction commits successfully. EndTransaction token with zero status is sent for the top-level transaction. -> The column value(s) retains the set value and the TPer mode, i.e. whether in MBR mode or not, and the behavior after a power cycle, change as the set value indicated.	
D5-1-3-1-2				Set() for a 'Enable' column or a 'Done' column, or for both, is invoked within a transaction. -> The column value(s) become the new value(s) that the Set() indicates immediately, but the mode of the TPer, i.e. whether in MBR mode or not, and the behavior after a power cycle, are not affected unless the transaction commits successfully. EndTransaction token with non-zero status is sent for the top-level transaction. -> The column value(s), the mode of the TPer and the behavior after a power cycle, back to the states before the transaction started.	

3.3.6. D6. MBR.Set() - Grammar and Effect -

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
D6. MBR.Set() - Grammar and Effect -					
		Effects of successful method invocation	(Description) The successful invocation of this method SHALL cause, (and a non-successful invocation SHALL NOT cause), the effects as below:		
D6-1-1-1-1				The value(s) of the bytes get set into the value the Set() indicates. So, if a Get() is issued to the rows, the Get() retrieves the value(s) the Set() indicated.	
D6-1-1-1-1(2)				If this method is invoked successfully outside of a transaction, following READ commands SHALL retrieve the value that this invocation indicated.	
D6-1-2-1-1		Effects of this method invocation within a transaction		Set() for the MBR table is invoked within a transaction. -> The row value(s) become the new value(s) that the Set() indicates immediately, but the data which is returned when READ commands addressing the MBR table during the MBR mode were issued, are not affected unless the transaction commits successfully. EndTransaction token with zero status is sent for the top-level transaction. -> The row value(s) retains the set value and the data read for READ commands change as the set value indicated.	
D6-1-2-1-2				Set() for the MBR table is invoked within a transaction. -> The row value(s) become the new value(s) that the Set() indicates immediately, but the data which is returned when READ commands addressing the MBR table during the MBR mode were issued, are not affected unless the transaction commits successfully. EndTransaction token with non-zero status is sent for the top-level transaction. -> The row value(s) and the data read for READ commands changes back to the values before the transaction started.	

3.3.7. D7. DataStore.Set() - Grammar and Effect -

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
D7. DataStore.Set() - Grammar and Effect -					
		Effects of successful method invocation	(Description) The successful invocation of this method SHALL cause, (and a non-successful invocation SHALL NOT cause), the effects as below:		
D7-1-1-1-1				The value(s) of the bytes get set into the value the Set() indicates. So, if a Get() is issued to the rows, the Get() SHALL retrieve the value(s) the Set() indicated.	
D7-1-2-1-1		Effects of this method invocation within a transaction		Set() for the DataStore table is invoked within a transaction. -> The row value(s) become the new value(s) that the Set() indicates immediately. EndTransaction token with zero status is sent for the top-level transaction. -> The row value(s) retains the set value.	
D7-1-2-1-2				Set() for the DataStore table is invoked within a transaction. -> The row value(s) become the new value(s) that the Set() indicates immediately. EndTransaction token with non-zero status is sent for the top-level transaction. -> The row value(s) back to the value(s) before the transaction started.	

3.3.8. D8. K_AES_*.GenKey () - Effect -

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
D8. GenKey() - Effect -					
D8-1-1-1-1	For all the objects in the K_AES_128/256 supported by the TPer	Grammar		If no parameters are encoded, the method invocation SHALL NOT fail because of that.	
		Effects of successful method invocation	(Description) The successful invocation of this method SHALL cause, (and a non-successful invocation SHALL NOT cause), the effects as below:		
D8-1-2-1-1				The media encryption key used to encrypt/decrypt user data changes.	
D8-1-3-1-1		Effects of this method invocation upon a transaction		GenKey() for a K_AES_XXX object is invoked within a transaction. -> The corresponding Locking Range's media encryption key is not affected unless the transaction committed successfully. EndTransaction token with zero status is sent for the top-level transaction. -> The Range's media encryption key changes. (XXX should be replaced with 128 or 256, depending on which, or both, table is supported by the TPer and upon which object GenKey was invoked.)	
D8-1-3-1-2				GenKey() for a K_AES_XXX object is invoked within a transaction. -> The corresponding Locking Range's media encryption key is not affected unless the transaction committed successfully. EndTransaction token with non-zero status is sent for the top-level transaction. -> The Range's media encryption key backs to the value before the transaction started. (XXX should be replaced with 128 or 256, depending on which, or both, table is supported by the TPer and upon which object GenKey was invoked.)	

3.3.9. D9. Activate() - Grammar and Effect -

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
D9. Activate() - Grammar and Effect -					
D9-1-1-1-1	For LockingSP.Activate()	Grammar		If no parameters are encoded, the method invocation SHALL NOT fail because of that.	
D9-1-2-1-1		Conditions		If the ATA security feature of the TPer is not enabled, the method SHALL NOT fail because of that.	
D9-1-2-1-2				If the ATA security feature of the TPer is enabled, the method SHALL fail with FAIL.	
		Effects of successful method invocation	(Description) The successful invocation of this method SHALL cause, (and a non-successful invocation SHALL NOT cause), the effects as below:		
D9-1-3-1-1			Common	For ATA devices, bit 1 of word 82, bit 1 of word 85 and all bits of word 89, 90, 92, 128 in the IDENTIFY DEVICE data SHALL be set to all-0.	
D9-1-3-1-2				LockingEnabled bit in Locking Feature Descriptor in the Level 0 Discovery Response data SHALL be set to 1.	
D9-1-3-1-3				The value of LifeCycleState column of the Locking SP object in the SP table SHALL be set to 0x09.	
D9-1-3-1-4				A startup of a session to the Locking SP can succeed.	
D9-1-3-1-5				All the data in the User-Addressable LBA Range's sectors SHALL NOT change.	
D9-1-3-1-6				(If the TPer supports RestrictedCommands table in the Locking SP) Each command described in the RestrictedCommands table in the Locking SP, issued with non-destructive value of Command Parameter (or CDB) and non-destructive data, if needed, SHALL succeed or SHALL be aborted as described in SIIS v1.00 depending on the value of the Allowed column of the corresponding object in the table.	
D9-1-3-2-1			If the Locking SP is in Manufactured-Inactive state	The value of PIN column of the Admin1 object in the C_PIN table, i.e. the PIN credential for the Admin1, in the Locking SP is the same as the SID credential.	
D9-1-3-3-1			If the Locking SP is in Manufactured state	The value of PIN column of the Admin1 object in the C_PIN table, i.e. the PIN credential for the Admin1, in the Locking SP does not change.	

3.3.10. D10. Revert()/RevertSP() - Grammar and Effect -

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
D10. Revert()/RevertSP() - Grammar and Effect -					
D10-1-1-1-1	For AdminSP.Revert()	Grammar		If no parameters are encoded, the method invocation SHALL NOT fail because of that.	
		Effects of successful method invocation	(Description) The successful invocation of this method SHALL cause, (and a non-successful invocation SHALL NOT cause), the effects as below:		
D10-1-2-1-1			Common	The session within which the AdminSP.Revert() was issued SHALL be aborted.	
D10-1-2-1-2				For ATA devices: i) if the device supports Manufactured-Inactive state of the Locking SP as its Original Factory State, and if the device has the ability to support ATA security feature set, and if the ATA security feature set is not disabled by Device Configuration Overlay or other methods, then bit 1 in word 82 in the IDENTIFY DEVICE data SHALL be set to 1, and bit 1 in word 85 and all bits in word 89, 90, 92 and 128 in the data SHALL be set to appropriate values, ii) otherwise, bit 1 in word 82, bit 1 in word 85 and all bits in word 89, 90, 92 and 128 in the IDENTIFY DEVICE data SHALL be set to 0.	
D10-1-2-1-3				i) If the device supports Manufactured-Inactive state of the Locking SP as its Original Factory State, LockingEnabled bit in Locking Feature Descriptor in the Level 0 Discovery Response data, SHALL be set to 0. ii) If the device supports Manufactured state of the Locking SP as its Original Factory State, LockingEnabled bit in Locking Feature Descriptor in the Level 0 Discovery Response data, SHALL be set to 1.	
D10-1-2-1-4				i) If the device supports Manufactured-Inactive state of the Locking SP as its Original Factory State, the value of LifeCycleState column of the Locking SP object in the SP table, SHALL be set to 0x08. ii) If the device supports Manufactured state of the Locking SP as its Original Factory State, the value of LifeCycleState column of the Locking SP object in the SP table, SHALL be set to 0x09.	
D10-1-2-1-5				i) If the device supports Manufactured-Inactive state of the Locking SP as its Original Factory State, a startup of a session to the Locking SP SHALL fail. ii) If the device supports Manufactured state of the Locking SP as its Original Factory State, a startup of a session to the Locking SP SHALL NOT fail because of that.	

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
D10-1-2-1-6				The value of PIN column of the SID object in the C_PIN table, i.e. the SID credential, in the Admin SP is set to the value of MSID credential.	
D10-1-2-1-7				(If the TPer supports RestrictedCommands table in the Locking SP) i) If the device supports Manufactured-Inactive state of the Locking SP as its Original Factory State, each command described in the RestrictedCommands table in the Locking SP, issued with non-destructive value of Command Parameter (or CDB) and non-destructive data, if needed, SHALL succeed or SHALL be aborted depending on the settings in the TPer other than the RestrictedCommands table. (* This is not to be tested because it hardly described the pass/fail criteria here.) ii) If the device supports Manufactured state of the Locking SP as its Original Factory State, each command described in the RestrictedCommands table in the Locking SP, issued with non-destructive value of Command Parameter (or CDB) and non-destructive data, if needed, SHALL succeed or SHALL be aborted depending on the value of the Allowed column of the corresponding object in the table.	
D10-1-2-2-1			If the Locking SP is in Manufactured state	All the data in the User-Addressable LBA Range's sectors SHALL be cryptographically erased.	
D10-1-2-2-2				All the values in the DataStore table SHALL be the value when the TPer is in its Original Factory State. (The row values in its Original Factory State are vendor-unique.)	
D10-1-2-2-3				All the values in the MBR table SHALL be the value when the TPer is in its Original Factory State. (The row values in its Original Factory State are vendor-unique.)	
D10-1-2-3-1			If the Locking SP is in Manufactured-Inactive state	All the data in the User-Addressable LBA Range's sectors SHALL NOT change.	
D10-2-1-1-1	For LockingSP.Revert()	Grammar		If no parameters are encoded, the method invocation SHALL NOT fail because of that.	
		Effects of successful method invocation	(Description) The successful invocation of this method SHALL cause, (and a non-successful invocation SHALL NOT cause), the effects as below:		
D10-2-2-1-1			Common	The session remains open.	

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
D10-2-2-1-2				For ATA devices: i) if the device supports Manufactured-Inactive state of the Locking SP as its Original Factory State, and if the device has the ability to support ATA security feature set, and if the ATA security feature set is not disabled by Device Configuration Overlay or other methods, then bit 1 in word 82 in the IDENTIFY DEVICE data SHALL be set to 1, and bit 1 in word 85 and all bits in word 89, 90, 92 and 128 in the data SHALL be set to appropriate values, ii) otherwise, bit 1 in word 82, bit 1 in word 85 and all bits in word 89, 90, 92 and 128 in the IDENTIFY DEVICE data SHALL be set to 0.	
D10-2-2-1-3				i) If the device supports Manufactured-Inactive state of the Locking SP as its Original Factory State, LockingEnabled bit in Locking Feature Descriptor in the Level 0 Discovery Response data, SHALL be set to 0. ii) If the device supports Manufactured state of the Locking SP as its Original Factory State, LockingEnabled bit in Locking Feature Descriptor in the Level 0 Discovery Response data, SHALL be set to 1.	
D10-2-2-1-4				i) If the device supports Manufactured-Inactive state of the Locking SP as its Original Factory State, the value of LifeCycleState column of the Locking SP object in the SP table, SHALL be set to 0x08. ii) If the device supports Manufactured state of the Locking SP as its Original Factory State, the value of LifeCycleState column of the Locking SP object in the SP table, SHALL be set to 0x09.	
D10-2-2-1-5				i) If the device supports Manufactured-Inactive state of the Locking SP as its Original Factory State, a startup of a session to the Locking SP SHALL fail. ii) If the device supports Manufactured state of the Locking SP as its Original Factory State, a startup of a session to the Locking SP SHALL NOT fail because of that.	
D10-2-2-1-6				(If the TPer supports RestrictedCommands table in the Locking SP) i) If the device supports Manufactured-Inactive state of the Locking SP as its Original Factory State, each command described in the RestrictedCommands table in the Locking SP, issued with non-destructive value of Command Parameter (or CDB) and non-destructive data, if needed, SHALL succeed or SHALL be aborted depending on the settings in the TPer other than the RestrictedCommnads table. (* This is not to be tested because it hardly described the pass/fail criteria here.) ii) If the device supports Manufactured state of the Locking SP as its Original Factory State, each command described in the RestrictedCommands table in the	

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
				Locking SP, issued with non-destructive value of Command Parameter (or CDB) and non-destructive data, if needed, SHALL succeed or SHALL be aborted depending on the value of the Allowed column of the corresponding object in the table.	
D10-2-2-2-1			If the Locking SP is in Manufactured state	All the data in the User-Addressable LBA Range's sectors SHALL be cryptographically erased.	
D10-2-2-2-2				All the values in the DataStore table SHALL be the value when the TPer is in its Original Factory State. (The row values in its Original Factory State are vendor-unique.)	
D10-2-2-2-3				All the values in the MBR table SHALL be the value when the TPer is in its Original Factory State. (The row values in its Original Factory State are vendor-unique.)	
D10-2-2-3-1			If the Locking SP is in Manufactured-Inactive state	All the data in the User-Addressable LBA Range's sectors SHALL NOT change.	
D10-2-2-3-2				All the values in the DataStore table SHALL be the value when the TPer is in its Original Factory State. (The row values in its Original Factory State are vendor-unique.)	
D10-2-2-3-3				All the values in the MBR table SHALL be the value when the TPer is in its Original Factory State. (The row values in its Original Factory State are vendor-unique.)	
D10-3-1-1-1	For LockingSP.RevertSP()	Grammar		If no parameters are encoded, the method invocation SHALL NOT fail because of that.	
D10-3-2-1-1		Condition		If the parameter 'KeepGlobalRangeKey' indicates 1, and if the Locking GlobalRange is read-unlocked or write-unlocked, the method invocation SHALL NOT fail because of that.	
D10-3-2-1-2				If the parameter 'KeepGlobalRangeKey' indicates 1, and if the Locking GlobalRange is both read-locked and write-locked, the method invocation SHALL fail with FAIL.	
		Effects of successful method invocation	(Description) The successful invocation of this method SHALL cause, (and a non-successful invocation SHALL NOT cause), the effects as below:		
D10-3-3-1-1			Common	The session within which the LockingSP.RevertSP() was issued SHALL be aborted.	

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
D10-3-3-1-2				For ATA devices: i) if the device supports Manufactured-Inactive state of the Locking SP as its Original Factory State, and if the device has the ability to support ATA security feature set, and if the ATA security feature set is not disabled by Device Configuration Overlay or other methods, then bit 1 in word 82 in the IDENTIFY DEVICE data SHALL be set to 1, and bit 1 in word 85 and all bits in word 89, 90, 92 and 128 in the data SHALL be set to appropriate values, ii) otherwise, bit 1 in word 82, bit 1 in word 85 and all bits in word 89, 90, 92 and 128 in the IDENTIFY DEVICE data SHALL be set to 0.	
D10-3-3-1-3				i) If the device supports Manufactured-Inactive state of the Locking SP as its Original Factory State, LockingEnabled bit in Locking Feature Descriptor in the Level 0 Discovery Response data, SHALL be set to 0. ii) If the device supports Manufactured state of the Locking SP as its Original Factory State, LockingEnabled bit in Locking Feature Descriptor in the Level 0 Discovery Response data, SHALL be set to 1.	
D10-3-3-1-4				i) If the device supports Manufactured-Inactive state of the Locking SP as its Original Factory State, the value of LifeCycleState column of the Locking SP object in the SP table, SHALL be set to 0x08. ii) If the device supports Manufactured state of the Locking SP as its Original Factory State, the value of LifeCycleState column of the Locking SP object in the SP table, SHALL be set to 0x09.	
D10-3-3-1-5				i) If the device supports Manufactured-Inactive state of the Locking SP as its Original Factory State, a startup of a session to the Locking SP SHALL fail. ii) If the device supports Manufactured state of the Locking SP as its Original Factory State, a startup of a session to the Locking SP SHALL NOT fail because of that.	
D10-3-3-1-6				(If the TPer supports RestrictedCommands table in the Locking SP) i) If the device supports Manufactured-Inactive state of the Locking SP as its Original Factory State, each command described in the RestrictedCommands table in the Locking SP, issued with non-destructive value of Command Parameter (or CDB) and non-destructive data, if needed, SHALL succeed or SHALL be aborted depending on the settings in the TPer other than the RestrictedCommands table. (* This is not to be tested because it hardly described the pass/fail criteria here.) ii) If the device supports Manufactured state of the Locking SP as its Original Factory State, each command described in the RestrictedCommands table in the	

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
				Locking SP, issued with non-destructive value of Command Parameter (or CDB) and non-destructive data, if needed, SHALL succeed or SHALL be aborted depending on the value of the Allowed column of the corresponding object in the table.	
D10-3-3-2-1			If the Locking SP is in Manufactured state	If the parameter 'KeepGlobalRangeKey' indicates 1, all the data in the sectors covered by the GlobalRange SHALL NOT change and the other data in the sectors covered by the ranges other than the GlobalRange SHALL be cryptographically erased.	
D10-3-3-2-3				If the parameter 'KeepGlobalRangeKey' indicates 0 or if the parameter is not encoded, all the data in the User-Addressable LBA Range's sectors SHALL be cryptographically erased.	
D10-3-3-2-4				Whichever value, true or false or not specified, was indicated as the value of the 'KeepGlobalRangeKey' parameter, all the values in the DataStore table SHALL be the value when the TPer is in its Original Factory State. (The row values in its Original Factory State are vendor-unique.)	
D10-3-3-2-5				Whichever value, true or false or not specified, was indicated as the value of the 'KeepGlobalRangeKey' parameter, all the values in the MBR table SHALL be the value when the TPer is in its Original Factory State. (The row values in its Original Factory State are vendor-unique.)	

3.3.11. D11. Power Cycle

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
D11. Power Cycle					
D11-1-1-1-1	C_PIN table	Tries column		If the value of the Persistence column of C_PIN table is set to False, a Power Cycle SHALL reset the value of the corresponding Tries column to 0.	
D11-2-1-1-1	RestrictedCommands table	Allowed column		(If the TPer supports RestrictedCommands table in the Locking SP) i) If Power Cycle is contained in the value of the AllowedTrueOnReset column of an object in the RestrictedCommands table, a Power Cycle SHALL set the value of the corresponding Allowed column to True. ii) If Power Cycle is contained in the value of the AllowedFalseOnReset column of an object in the RestrictedCommands table, a Power Cycle SHALL set the value of the corresponding Allowed column to False. iii) If Power Cycle is not contained in the value of the AllowedTrueOnReset column nor in the value of the AllowedFalseOnReset column of an object in the RestrictedCommands table, a Power Cycle SHALL NOT affect the value of the corresponding Allowed column.	

4. Annex A Interface Specific Tests

4.1. ATA specific Tests

4.1.1. A0. IDENTIFY DEVICE

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
A0. IDENTIFY DEVICE					
A0-1-1-1-1	IDENTIFY DEVICE	Word 48		For ATA device, the Word48 bit 0 in the IDENTIFY DEVICE data of the TPer SHALL be set to 1.	
A0-1-1-1-2		Word 119		For ATA device, if Word 119 bit 6 in the IDENTIFY DEVICE data of the TPer is set to 1, verify that SET FEATURES with subcommand code 0xC3, and bit 0 of the Count field set to 1b succeed. If it succeeded, the TPer supports Sense Data Reporting feature and enabled it. If it failed, the TPer doesn't support Sense Data Reporting feature. Some tests in this document SHALL follow that fact.	

4.1.2. A1. TRUSTED SEND / TRUSTED RECEIVE

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
A1. TRUSTED SEND / TRUSTED RECEIVE					
A1-1-1-1-1	TRUSTED SEND and TRUSTED SEND DMA, basic tests	Transfer Length		If the Transfer Length in a TRUSTED SEND (as well as TRUSTED SEND DMA) is set to 0, the Security Protocol is set to 0 or 1 and the Security Protocol Specific is set to one of the TPer's static ComIDs, then the data transfer SHALL NOT occur and the command SHALL be aborted at the ATA interface level with: i) ST = 0x51 and ER = 0x04 or ii) ST has bit 1 set, Sense Key = ILLEGAL REQUEST and ASC/ASCQ = INVALID FIELD IN CDB if the TPer supports Sense Data Reporting feature set and it is enabled.	
A1-1-1-1-2				If the Transfer Length in a TRUSTED SEND (as well as TRUSTED SEND DMA) is greater than the value of the TPer's MaxComPacketSize property, the Security Protocol is set to 1 and the Security Protocol Specific is set to one of the TPer's static ComIDs, then the data transfer SHALL NOT occur and the command SHALL be aborted at the ATA interface level with: i) ST = 0x51 and ER = 0x04 or ii) ST has bit 1 set, Sense Key = ILLEGAL REQUEST and ASC/ASCQ = INVALID FIELD IN CDB if the TPer supports Sense Data Reporting feature set and it is enabled.	
A1-1-3-1-3		Security Protocol Specific	When the requested Security Protocol = 1 and the requested Transfer Length (in 512-byte blocks) = any value between 1 and the one the TPer's MaxComPacketSize property for the ComID in the command parameter	If the TPer is not in the Awaiting IF-SEND state for the ComID in a TRUSTED SEND (as well as TRUSTED SEND DMA), then the data transfer SHALL NOT occur and the command SHALL be aborted at the ATA interface level with: i) ST = 0x51 and ER = 0x04 or ii) ST has bit 1 set, Sense Key = ILLEGAL REQUEST and ASC/ASCQ = COMMAND SEQUENCE ERROR if the TPer supports Sense Data Reporting feature set and it is enabled.	
A1-1-3-1-4				If the Security Protocol Specific in a TRUSTED SEND (as well as TRUSTED SEND DMA) is set to one of the values of the non reserved range(0x1000-0xFFFF), but is one of the TPer's inactive ComIDs, then: i) if the TPer supports dynamic ComID allocation, the data transfer SHALL be completed successfully and the command SHALL succeed with ST = 0x50, but the payload SHALL be discarded by the TPer. ii) if the TPer does not support dynamic ComID allocation, ii-a) the TPer SHALL perform the same behaviors described in i) or ii-b) the data transfer SHALL NOT occur and the command SHALL be aborted at the ATA interface level with: ii-b-1) ST = 0x51 and ER = 0x04 or ii-b-2) ST has bit 1 set, Sense Key = ILLEGAL REQUEST and ASC/ASCQ = INVALID FIELD IN CDB if the TPer	

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
				supports Sense Data Reporting feature set and it is enabled.	
A1-1-3-1-5				If the Security Protocol Specific in a TRUSTED SEND (as well as TRUSTED SEND DMA) is set to one of the values of the reserved range(0x0000-0x0FFF), but is one of the TPer's unsupported ComIDs, then the data transfer SHALL NOT occur and the command SHALL be aborted at the ATA interface level with: i) ST = 0x51 and ER = 0x04 or ii) ST has bit 1 set, Sense Key = ILLEGAL REQUEST and ASC/ASCQ = INVALID FIELD IN CDB if the TPer supports Sense Data Reporting feature set and it is enabled.	
A1-2-1-1-1	TRUSTED RECEIVE and TRUSTED RECEIVE DMA, basic tests	Transfer Length		If the Transfer Length in a TRUSTED RECEIVE (as well as TRUSTED RECEIVE DMA) is set to 0, then the data transfer SHALL NOT occur and the command SHALL be aborted at the ATA interface level with: i) ST = 0x51 and ER = 0x04 or ii) ST has bit 1 set, Sense Key = ILLEGAL REQUEST and ASC/ASCQ = INVALID FIELD IN CDB if the TPer supports Sense Data Reporting feature set and it is enabled.	
A1-2-3-2-2		Security Protocol Specific	When the requested Security Protocol = 1 and the requested Transfer Length (in 512-byte blocks) = any value between 1 and the one the Host's MaxComPacketSize, and less than or equal to the TPer's MaxResponseComPacketSize property for the ComID in the command parameter	If the Security Protocol Specific in a TRUSTED RECEIVE (as well as TRUSTED RECEIVE DMA) is set to one of the values of the non reserved range(0x1000-0xFFFF), but one of the TPer's inactive ComIDs, then: i) if the TPer supports dynamic ComID allocation, the data transfer SHALL be completed successfully and the command SHALL succeed with ST = 0x50, but in the payload, the ComID field in the ComPacket header = the requested ComID, ComID Extension field = 0xFFFF, and other fields in the ComPacket header = all-0s. ii) if the TPer does not support dynamic ComID allocation, ii-a) the TPer SHALL perform the same behaviors described in i) or ii-b) the data transfer SHALL NOT occur and the command SHALL be aborted at the ATA interface level with: ii-b-1) ST = 0x51 and ER = 0x04 or ii-b-2) ST has bit 1 set, Sense Key = ILLEGAL REQUEST and ASC/ASCQ = INVALID FIELD IN CDB if the TPer supports Sense Data Reporting feature set and it is enabled.	

ID	Item				
	Group	Subgroup	Subsubgroup	Detail	Memo
A1-2-3-2-3				If the Security Protocol Specific in a TRUSTED RECEIVE (as well as TRUSTED RECEIVE DMA) is set to one of the values of the reserved range(0x0000-0x0FFF), but one of the TPer's unsupported ComIDs, then the data transfer SHALL NOT occur and the command SHALL be aborted at the ATA interface level with: i) ST = 0x51 and ER = 0x04 or ii) ST has bit 1 set, Sense Key = ILLEGAL REQUEST and ASC/ASCQ = INVALID FIELD IN CDB if the TPer supports Sense Data Reporting feature set and it is enabled.	