ENDL
TEXAS

Date: 20 August 2003
To: T10 Technical Committee & SNIA OSD TWG
From: Ralph O. Weber
Subject: Review of SNIA OSD TWG comments on OSD r07a

This proposal contains the lodged against OSD r07a by members of the SNIA OSD Technical Working Group as well as the resolutions for these comments to be applied to OSD r08.

The purpose of this proposal is to document the requested changes and track the discussion of making those changes.

All references to OSD pages are references to printed page numbers (not PDF page numbers) osd-r07a.pdf.

The number in square brackets at the end of each comment description counts all the comments discussed in this document.

Revision History

r0 All comments that I could find are present. Many of the comments are unprocessed.

Using this document

In this document, all references to other comments are PDF hot links. This includes the references in the several lists that follow this page.

The PDF bookmarks organize comments by comments contributor and order that the comment was received. The lists following this page organize the comments by disposition:

- Unresolved
- Rejected
- Comments Requiring No Action (e.g., comments that ask questions)
- Deferred to OSD-2
- Technical Comments (that will result in substantive changes to OSD)
  - Accepted exactly as proposed
  - Accepted with changes (and said changes may totally reverse the sense of the original comment)
- Editorial Comments
  - Accepted exactly as proposed
  - Accepted with changes (and said changes may totally reverse the sense of the original comment)

Persons submitting comments should review the lists following this page as a quick way to verify satisfactory disposition of their comments.

Unresolved Comments List

Editor 4) LIST command Index and Sort Order parameters 11
EMC 1) Encrypted connection required for Credential transmittal 13
EMC 4) Describe attacks, threats, and risks protected against by security levels 13
EMC 5) Extensible hash and encryption functions required 13
IBM 1) Incorporate Security Terms in Glossary 15
IBM 2) Should 'Overall Architecture' mention security? 15
IBM 4) Various Security Manager requirements 15
IBM 5) 'command key' s/b 'capability key' 15
IBM 6) Security Level 1 definition incorrect 15
IBM 24) Remove the WK_OBJ bit from the capability definition 19
Panasas 1) “Create Attributes Page” command and Proposed Attribute Templates 22
Panasas 3) Atomicity of Set Attributes writes 23
Panasas 4) Attribute Directories 23
Panasas 5) Size of Object Attribute Name Space 24
Panasas 7) Get Attributes Parameters 25
Panasas 9) Security Attributes Page 25
Panasas 10) Create command and the Create and Write command 25
Panasas 11) Root object “Used Capacity” attribute 26
Panasas 12) Root object “OSD Security Level” attribute 26
Panasas 13) Group and User-object Information Attributes Page 26
Panasas 14) Command with GetAttr and/or SetAttr and execution ordering 26
Panasas 15) Size of object address space 27
Panasas 16) Last-Access time 27
Panasas 17) Group Resources Attribute “Remaining Capacity” 27
Panasas 18) Terminology 28
Panasas 20) Fields important to include in the Capability 30
Panasas 21) Capabilities Permission Bits 30
Panasas 22) Key Version field too large 31
Panasas 23) Timestamp bypass 31
Panasas 24) Credential Creation Time 31
Panasas 25) Credential Nonce 31
Seagate 1) Attribute inheritance wording is unclear 32
Seagate 2) Last attribute is FFFF FFFEh 32
Seagate 3) Attribute directories are irrelevant 32
Seagate 4) Why limit sessions to READ, WRITE, and APPEND? 32
Seagate 6) Terminology inconsistent between Figure 4 and text 32
Seagate 10) Level 4 overview sentence does not make sense 33
Seagate 14) Example needed in the OSD model 34
Seagate 19) Remove the CREATE ATTRIBUTES PAGE command 35
Seagate 21) IMPORT USER OBJECT CDB needs more fields 35
Seagate 23) Add Session Template to OPEN 35
Seagate 24) Reduce WRITE byte count to 32 bits 35
Seagate 25) Increase Object Logical Length to 96 bits 36
Seagate 26) Limit access available to list format attributes 36
Seagate 27) Remove directory pages and definitions 36
Seagate 28) Root 'Used Capacity' attribute should count all user objects 36
Seagate 32) Does User Object Used Capacity report space actually used? 37
Seagate 33) Increase Group Count size from 4 to 8 bytes 37
Seagate 35) Move 'Object Logical Length' to User Object Info Page 37

Rejected Comments List

IBM 3) Security Manager should not be optional . . . 15
IBM 11) Request more Additional Sense Code assignments . . . 16
IBM 13) Request more Additional Sense Code assignments . . . 17
IBM 15) Request more Additional Sense Code assignments . . . 17
IBM 23) What is the 'No Credential' credential format? . . . 19
IBM 25) Request more Additional Sense Code assignments . . . 19
Seagate 12) Change 'device server' to 'OSD device'. . . . 33
Seagate 13) Keep subclause 4.7.2 (Discovery and Configuration) . . . 34
Seagate 36) Mention of Root Remaining Capacity is redundant . . . 37

No Action Requested, No Action Taken Comments List

IBM 12) Is 'Security Token' another name for 'ChannelID' 17
IBM 21) Illegal CDB truncation over defined 19
IBM 27) Where is Data-In Integrity Check Value 20
Seagate 15) Linked Command Support. 34
Seagate 38) Why return session id in the Current Command attributes page?. 38

Comments With Implementation Deferred to OSD-2

Panasas 6) Large Capability and Where To Store It 24
Seagate 20) Define FLUSH PARTITION & FLUSH OSD 35

Substantive Comments Accepted As Proposed

Substantive Comments Accepted With Noted Changes

OSD TWG 1) Incorporate 03-278r0 ObS Identifying Objects 10
OSD TWG 2) Incorporate 03-279r0 Object Store Security Document. 10
OSD TWG 3) Incorporate 03-280r1 OSD Grouping and Attributes 10
Editor 2) Remove subclause 4.7.2 (Discovery and Configuration) 11
EMC 6) Nonce requirements inadequate. 13
IBM 7) Recombine security Levels 1 and 2 as per the SNIA OSD TWG description. 16
IBM 14) Level 3 (2) does not Response Integrity Check Value 17
IBM 16) Level 4 does not Response Integrity Check Value. 18
IBM 20) CDBs are not encrypted 18
IBM 22) Credential format does not match Security Document. 19
IBM 28) Move OSD Security attribute to the Group Information attributes page 20
IBM 29) CDB Integrity Check Values are only 12 bytes. 20
Panasas 8) Get and Set Attribute Parameters. 25
Panasas 19) Credential Format 29
Seagate 16) Get/Set Parameters Changes 34
Seagate 22) Increase Index field size in LIST CDB. 35
Seagate 29) Add Security Level to Group attributes 36
Seagate 34) ’total capacity' s/b 'remaining capacity' 37
Seagate 39) Update Attributes Lists definitions for restricted access 38

Accepted As Proposed Non-Substantive Comments List

Editor 1) Add model cross references to security data field definitions . . . 11
Editor 3) FLUSH OBJECT and FORMAT OSD not in alphabetical order . . . 11
Editor 7) Remove Annexes B, C, and D . . . 12
EMC 2) Add cross reference to Credential Format . . . 13
EMC 3) 'Digital Signature' s/b 'Integrity Check Value' . . . 13
IBM 10) Missing 'is' . . . 16
Seagate 5) 'a' s/b 'an' . . . 32
Seagate 7) Remove 'that' & insert 'is' . . . 33
Seagate 8) Too many 'may's . . . 33
Seagate 18) Last Byte number wrong in CREATE AND WRITE format . . . 34

Accepted With Noted Changes Non-Substantive Comments List

IBM 9) Too many 'may's. 16
IBM 18) Term 'Digital Signature' is incorrect 18
IBM 30) Eliminate 'Digital Signature' 20
Seagate 9) Security Level 1 1st sentence is nonsense 33
Seagate 11) Insert cross reference to table 15 33
Seagate 30) Wrong attribute number for Group Username. 36
Seagate 31) Wrong attribute number for User Object Username 36

1. SNIA OSD TWG

The following requests have been made on behalf of sub-groups within the SNIA OSD TWG.

OSD TWG 1) Incorporate 03-278r0 ObS Identifying Objects (Accepted, Substantive) [1]
Global

Incorporate 03-278r0 (ObS Identifying Objects), a description of enhancements to the identification of OSD objects.

Editor’s Notes: The following differences between 03-278r0 and what has been incorporated in OSD r08 are worthy of note:

1) Sections 2 (Requirements), 3 (Architectural assumptions), and … have not been incorporated because they contain explanatory material that is not appropriate for inclusion in a T10 standard.
2) …

OSD TWG 2) Incorporate 03-279r0 Object Store Security Document (Accepted, Substantive) [2]
Global

Incorporate 03-279r0 (Object Store Security Document), a description of security features that may be provided by OSD devices.

Editor’s Notes: The following differences between 03-279r0 and what has been incorporated in OSD r08 are worthy of note:

1) Sections 1 (Revision History), and … have not been incorporated because they contain explanatory material that is not appropriate for inclusion in a T10 standard.
2) …

OSD TWG 3) Incorporate 03-280r1 OSD Grouping and Attributes (Accepted, Substantive) [3]
Global

Incorporate 03-280r1 (OSD Grouping and Attributes), a description of group and attributes features that may be provided by OSD devices.

Editor’s Notes: The following differences between 03-280r1 and what has been incorporated in OSD r08 are worthy of note:

1) Sections 0 (Revision History), 1 (Introduction), and … have not been incorporated because they contain explanatory material that is not appropriate for inclusion in a T10 standard.
2) …

2. Editor Changes in Preparation for T10 Letter Ballot

The following changes have been made by the editor in preparation for the anticipated T10 Letter Ballot.

Editor 1) Add model cross references to security data field definitions (Accepted, Editorial) [4]
pages 24-25, 4.6.5.2 Global

For each security parameter are response value, add a cross reference to the subclause where the field or attribute containing that value is defined.

Editor 2) Remove subclause 4.7.2 (Discovery and Configuration) (Accepted, Substantive) [5]
Page 28, 4.7.2

Subclause 4.7.2 titled "Startup — discovery and configuration" is not appropriate content for a SCSI standard. Nowhere in SCSI is the method by which SCSI devices are discovered defined. The opinion of T10 (as I understand it) is that device discovery is a function preformed by unique features of each SCSI Transport Protocol (e.g., the Fiber Channel Name Server).

Furthermore, subclause 4.7.2 calls upon OSD device server to perform functions not defined in the remainder of the OSD working draft, e.g., 'the OSD device shall identify itself to all initiators'. As if the absence of definitions for OSD commands needed to accomplish this were not bad enough, performing this function in the context of the OSD command set would require switching initiator and target roles since only initiators can send unsolicited messages and the OSD device server is a component of a target.

Subclause 4.7.2 will be removed.

Editor 3) FLUSH OBJECT and FORMAT OSD not in alphabetical order (Accepted, Editorial) [6]
page 46, 6.1, table 33 & pages 59-62, 6.8 & 6.9

The FLUSH OBJECT and FORMAT OSD commands need to appear in the order shown here.

Editor 4) LIST command Index and Sort Order parameters (Unresolved) [7]

page 66-67, 6.12

There is an ongoing debate about what to do with the Index and Sort Order parameters.

In the absence of any clear agreement in the SNIA OSW TWG, the following changes will be made:

a) The Index field size will be changed to match the field size of an object identifier,
b) The coded values for Sort Order will be modified to indicate that support for Vendor Specific sort ordering is mandatory while support for other sort ordering choices is optional,
c) Specific wording will be added to indicate that a LIST command containing an unsupported Sort Order value shall be terminated with a CHECK CONDITION status, and
d) Specific wording will be added to indicate that a LIST command containing a Vendor Specific Sort Order and a non-zero Index value shall be terminated with a CHECK CONDITION status.

Editor 5) REMOVE [object] should update Group Modification time (Accepted, Substantive) [8]
page 97, 7.1.2.13, last p on pg
see also: comment Seagate 37)

Shouldn’t the data modified time be updated when a user object is removed, too?

Editor 6) Add sense data to response integrity check value (Accepted, Substantive) [9]
page 101, 7.1.2.15

Change from:

~~If the OSD security level is 0 or 1, the response digital signature attribute (number 4h) shall contain zero. Otherwise, the response digital signature attribute shall contain a digital signature (see 4.6.5.5) that is computed using the command key (see 4.6.5.1) and covering the following data:~~

~~a) If the OSD security level is 2, the covered data shall be a security token that the service delivery subsystem returns only to the application client and the device server; or~~
~~b) If the OSD security level is 3 or greater, the covered data shall be the status code returned for the current command plus the contents of the response nonce attribute in the Current Command attributes page (i.e., this attributes page).~~

to:

If the OSD security level is 0 or 1, the response digital signature attribute (number 4h) shall contain zero. Otherwise, the response digital signature attribute shall contain a digital signature (see 4.6.5.5) that is computed using the command key (see 4.6.5.1) and covering the following data:

a) If the OSD security level is 2, the covered data shall be a security token that the service delivery subsystem returns only to the application client and the device server concatenated with the contents of the response nonce attribute in the Current Command attributes page (i.e., this attributes page); or
b) If the OSD security level is 3 or greater, the covered data shall be the concatenation of:
   1) The status code returned for the current command;
   2) If the status is GOOD, the sense data returned for the current command; and
   3) The contents of the response nonce attribute in the Current Command attributes page.

Editor 7) Remove Annexes B, C, and D (Accepted, Editorial) [10]

The following Annexes have been removed because that are not a normative part of the OSD standard and because their content serves to confuse readers regarding implementation requirements:

Annex B — Research Notes
Annex C — OSD Related Topics
Annex D — Known Unresolved Issues or Uncompleted Topics

Since these annexes were never normative (i.e., they have always been labeled 'Informative') this change is deemed not substantive.

3. EMC Corp.

David Black from EMC Corp. submitted the following comments.

EMC 1) Encrypted connection required for Credential transmittal (Unresolved) [11]

page 23, 4.6.5.1
see also: comment IBM 4)

> The protocol between the application client and the security manager is not defined by this standard; however, the structure of the credential returned from the security manager to the application client is.

That protocol must be capable of encrypting the credential's command key, as secrecy of the command key is used to establish that usage of the credential is valid, and to protect command and data integrity. This is needed for level 2 and up.

EMC 2) Add cross reference to Credential Format (Accepted, Editorial) [12]

pages 22-27, 4.6.5

This clause could use a pointer to the Credential Format defined in 5.1.2.1.

EMC 3) 'Digital Signature' s/b 'Integrity Check Value' (Accepted, Editorial) [13]

Global

The term "digital signature" is misused in the standard, because a symmetric (single key) crypto system is being used. Encrypting a hash with a key from a symmetric crypto system is usually not considered a digital signature. This is fairly important, as most security-aware readers will associate the term "digital signature" with the computational overhead of asymmetric (public- key) crypto. Use something like "integrity check value".

EMC 4) Describe attacks, threats, and risks protected against by security levels (Unresolved) [14]

pages 24-25, 4.6.5.2

The security levels describe what functions are performed, but do not completely describe what attacks/threats/risks they protect against. For example, it is unclear whether level 3 is intended to prevent replay attacks. Some summary text describing the various threats and the appropriate levels needed to counter the threats would be quite useful.

EMC 5) Extensible hash and encryption functions required (Unresolved) [15]

page 27, 4.6.5.5

Table 12 is bad news - the hash and encryption functions are fixed, implicitly selected, and can't be changed. They need to be explicit, extensible and indicated in the credential somehow (e.g., make sure that AES can be added in addition to 3DES). It might be sufficient to say that changes to these functions can (only) be made by changing the credential format value.

EMC 6) Nonce requirements inadequate (Accepted, Substantive) [16]

page 26, 4.6.5.4

> When OSD security levels that employ nonces are in effect, recipients of nonces (i.e., device severs and applications clients) shall maintain lists of recently received nonce values. The number of entries in these lists is vendor specific.

Not good enough - need to specify a minimum number of entries. Between this, and the loose language specifying the virtual timestamp incrementing requirements (The number of nonce values containing the same virtual timestamp should be less than 1 000), the current spec is probably vulnerable to replay attacks.

Editor’s Note: This comment will be resoved as as described in comment IBM 17) and comment OSD TWG 2).

4. IBM

Michael Factor, Dalit Naor, and Julian Satran from IBM submitted the following comments.

IBM 1) Incorporate Security Terms in Glossary (Unresolved) [17]

pages 5-8, 3.1 & 3.2
see also: comment Panasas 18)

Section 3.1.5 needs to be filled in; also should add other definitions from security document to this section

IBM 2) Should 'Overall Architecture' mention security? (Unresolved) [18]

pages 12-13, 4.2

Should this section mention security?

IBM 3) Security Manager should not be optional (Rejected) [19]

page 13, 4.3, a,b,c list

I don't know if we want to say that the security manager is an optional constituent. I believe its function, constructing the credentials, must be provided unless we are running with no security.

Reason for Rejection: So long as there is an option for running without security, the security manager is optional because its presence is inappropriate when security is not enabled.

IBM 4) Various Security Manager requirements (Unresolved) [20]

page 13, 4.3, last p on pg 13
see also: comment EMC 1)

The requirement of encryption on the channel between the ObS and security manager is too strong; it turns out that the requirements on this channel are the same as the requirements on the channel between Object stores and hosts. In this same paragraph, you state that the Security manager can reside in the OSD storage device or initiator; it can also reside on another system.

IBM 5) 'command key' s/b 'capability key' (Unresolved) [21]

Global

In section 4.6.5.1 in the paragraph under figure 4 [and in numerous other locations throughout out the draft] you refer to a command key. This should be a CAP_Key or capability key for consistency.

IBM 6) Security Level 1 definition incorrect (Unresolved) [22]

page 23, 4.6.5.1, last p on pg

Individual commands are not signed with level 1 security (i.e., integrity of the capability)

IBM 7) Recombine security Levels 1 and 2 as per the SNIA OSD TWG description (Accepted, Substantive) [23]

pages 24-25, 4.6.5.2
see also: comment OSD TWG 2) and comment Seagate 9)

In section 4.6.5.2, there is an extra security level since integrity of credential and ownership of the credential were split. I don't think these two levels should not be split. What was the reason you split them?

Editor's Note: The splitting of these security levels will be removed and as part of that work the editorial comment described in comment Seagate 9) will be resolved.

I split them based on a confusion over the communication steps in which the Credential Integrity Check Value needs to be transferred. See comment IBM 31) and comment IBM 32) for a discussion of the errors resulting from this confusion.

Also, past discussions have indicated that a way was desired for initiators to prepare unsecured Capability values that could somehow be validated by the OSD. I was trying to preserve that function too.

IBM 8) Security enforcement is per partition (Accepted, Substantive) [24]

page 24, 4.6.5.2, last p in clause

The security level is enforced on a per partition basis.

IBM 9) Too many 'may's (Accepted, Editorial) [25]

page 24, 4.6.5.2.1
same problem identified in comment Seagate 8)

The first sentence is unclear.

Editor's Note: Change from:

> An OSD may request may be made without including any of the OSD security model features.

to:

> An OSD ~~may~~ request may be made without including any of the OSD security model features.

IBM 10) Missing 'is' (Accepted, Editorial) [26]

page 24, 4.6.5.2.2, p 2, s1

Missing the word "is" after "credential".

IBM 11) Request more Additional Sense Code assignments (Rejected) [27]

page 24, 4.6.5.2.2

Can we be more specific on error reasons; in the updated version of the security document, I will have a draft list of error reasons.

Reason for Rejection: The sense key specified for the error described in this section is ILLEGAL REQUEST. As described in SPC-3 r14 subclause 4.5.2.4.1 (page 31), the ILLEGAL REQUEST sense key indicates that the Sense Key Specific sense data field (or descriptor) may contain pointers to the exact CDB byte and bit determined to be in error. In this case, the field is a Integrity Check Value contained in the CDB. Therefore, the ability to identify this specific field in error is already provided by SCSI.

IBM 12) Is 'Security Token' another name for 'ChannelID' (No Action) [28]
page 24, 4.6.5.2.3, p 2

Is the "security token" what we had referred to as a "ChannelID"?

Editor's Note: Yes. The word 'Channel' carries too much T10 baggage to be used in the OSD working draft.

IBM 13) Request more Additional Sense Code assignments (Rejected) [29]
page 25, 4.6.5.2.4

Can we be more specific on error reasons.

Reason for Rejection: The sense key specified for the error described in this section is ILLEGAL REQUEST. As described in SPC-3 r14 subclause 4.5.2.4.1 (page 31), the ILLEGAL REQUEST sense key indicates that the Sense Key Specific sense data field (or descriptor) may contain pointers to the exact CDB byte and bit determined to be in error. In this case, the field is the CDB Integrity Check Values contained in the CDB. Therefore, the ability to identify this specific field in error is already provided by SCSI.

IBM 14) Level 3 (2) does not Response Integrity Check Value (Accepted, Substantive) [30]
page 25, 4.6.5.2.4
see also: comment Editor 6)

Where does the MAC for the response get described?

Editor's Note: At the end of the subclause add the following:

The device server constructs an integrity check value covering:

a) The status byte; and
b) If the status is CHECK CONDITION, the sense data.

The application client validates the integrity check value.

IBM 15) Request more Additional Sense Code assignments (Rejected) [31]
page 25, 4.6.5.2.5

Can we be more specific on error reasons.

Reason for Rejection: The sense key specified for the error described in this section is ILLEGAL REQUEST. As described in SPC-3 r14 subclause 4.5.2.4.1 (page 31), the ILLEGAL REQUEST sense key indicates that the Sense Key Specific sense data field (or descriptor) may contain pointers to the exact CDB byte and bit determined to be in error. In this case, the field is one of several Integrity Check Values contained in the CDB. Therefore, the ability to identify this specific field in error is already provided by SCSI.

IBM 16) Level 4 does not Response Integrity Check Value (Accepted, Substantive) [32]
page 25, 4.6.5.2.5
see also: comment Editor 6)

Where does the MAC for the status response get described?

Editor's Note: Replace:

> ~~Also, the device server constructs a digital signature for the contents of the Data-In Buffer using the command key and the application client validates the digital signature.~~

with:

> The device server constructs the following integrity check values using the command key covering:
>
> a) The following response bytes:
>   A) The status byte; and
>   B) If the status is CHECK CONDITION, the sense data.
>
> and
>
> b) The Data-In Buffer.
>
> The application client validates the integrity check values.

IBM 17) Update description of Nonces as per latest Security document (Accepted, Substantive) [33]
page 26, 4.6.5.4
see also: comment OSD TWG 2) and comment EMC 6)

I believe this section will need to be updated to reflect the conclusions we reached regarding nonces. This will appear in the updated security document. In particular, I think we need to further specify the format and the rules for accepting a nonce (e.g., rule b in this section is not a necessary condition for rejecting a nonce). Note most of the text in the security document regarding nonces should probably be treated as explanatory

IBM 18) Term 'Digital Signature' is incorrect (Accepted, Editorial) [34]
Global & page 27, 4.6.5.5
see also: comment IBM 30)

A digital signature is not necessarily an "encrypted value" and in particular it is not an encrypted value when we calculated it with a MAC.

Editor's Note: As described in the response to comment EMC 3), the term 'Digital Signature' will be replaced globally with 'Integrity Check Value'.

IBM 19) No Encryption in Integrity Check Value (Accepted, Substantive) [35]
page 27, 4.6.5.5

The description of preparation of a digital signature is incorrect -- we only require step 1 and not step 2. Also in table 12, we do not use encryption at all in the protocol

IBM 20) CDBs are not encrypted (Accepted, Substantive) [36]
page 32, 5.1.1, p 1

No encryption

Editor's Note: The first paragraph in 5.1.1 will be removed in its entirety.

IBM 21) Illegal CDB truncation over defined (No Action) [37]
page 33, 5.1.1, a,b list

Isn't bullet b) a special case of bullet a)

Editor's Note: In the absence of bullet b), the standard might be construed to allow after the PERMISSIONS BIT MASK field. This seems highly undesirable.

IBM 22) Credential format does not match Security Document (Accepted, Substantive) [38]
page 35, 5.1.2.1, Table 20

There are lots of differences in the fields and their sizes between the description in the security document and this table. I hope I have the document sufficiently precise at this point although I realize some of the differences are simply moving information.

Editor's Note: This comment will be resolved as described in the response to comment OSD TWG 2).

IBM 23) What is the 'No Credential' credential format? (Rejected) [39]
page 36, 5.1.2.1, Table 21

Why is no credential defined as a credential format? I don't think this is needed given we can have a no security level. I think we might, however, define a credential of all zeros to be equivalent to no credential.

Editor's Note: Checking 100+ bytes to be sure that they are all zero seems like an unwarrantable burden to detect the absence of a credential (aka capability). Checking a positively defined code value in a single byte seems more practical. Furthermore, when no credential (aka capability) is present, the 100+ bytes following the coded value need not be included in the CDB.

IBM 24) Remove the WK_OBJ bit from the capability definition (Unresolved) [40]

page 36, 5.1.2.1

I don't understand the WK_OBJ and its use in security -- where is this from?

IBM 25) Request more Additional Sense Code assignments (Rejected) [41]
pages 34-44, global in 5.1.2

In many places: I think we need to give more precise error indications

Reason for Rejection: For every CHECK CONDITION status described in subclause 5.1.2, the sense key specified for the error described in this section is ILLEGAL REQUEST. As described in SPC-3 r14 subclause 4.5.2.4.1 (page 31), the ILLEGAL REQUEST sense key indicates that the Sense Key Specific sense data field (or descriptor) may contain pointers to the exact CDB byte and bit determined to be in error. Therefore, the ability to identify this specific field in error is already provided by SCSI.

IBM 26) Capabilities defined for whole objects only (Accepted, Substantive) [42]
page 38, 5.1.2.1.2

We have decided that in the first version we will only support credentials for whole objects. I believe this section needs to be simplified accordingly

IBM 27) Where is Data-In Integrity Check Value (No Action) [43]
page 39, 5.1.2.2

Is there a corresponding Data-In digital signature that is needed?

Editor's Note: The Data-In Integrity Check Value is not transferred in the CDB because it must be transferred from the target to the initiator, whereas the CDB is transferred in the opposite direction. The definition of the Data-In Integrity Check Value can be found in 7.1.2.15 on page 101.

IBM 28) Move OSD Security attribute to the Group Information attributes page (Accepted, Substantive) [44]
page 87, 7.1.2.6
see also: comment Seagate 29)

Is the root object defined per partition -- the security level needs to be defined per partition

Editor's Note: This comment will be resolved as described in the response to comment Panasas 12).

IBM 29) CDB Integrity Check Values are only 12 bytes (Accepted, Substantive) [45]
page 34, 5.1.2, table 19

Bytes 44-63 and bytes 76- 95 should be 12 bytes only (instead of 20) since here we can use the truncated mode of HMAC-SHA1.

Editor's Note: In keeping with the requested change, the sizes of the Response Digital Signature (aka Integrity Check Value) and Data-In Integrity Check value will be changed from 20 to 12.

IBM 30) Eliminate 'Digital Signature' (Accepted, Editorial) [46]
Global
see also: comment IBM 18)

This is a general comment on the draft that has to do mainly with nomenclature but I believe is important. Section 4.6.5.5 defines a 'digital signature' in a manner closer to that defined in the crypto literature (based on public keys). However, our protocol does not use these type of digital signatures. The protocol uses only a MAC computation (based on symmetric keys) which has the unreversability property and which authenticates the data; it certainly does not encrypt the data but merely the use of the term 'digital signature' can be misleading here.

Editor's Note: Throughout the term 'Digital Signature' will be replaced with 'Integrity Check Value' as described in comment EMC 3).

IBM 31) Only Capability is sent to OSD in the CDB (Accepted, Substantive) [47]
Global

The client sends to the device only the capability, and the capability contains all the fields of Table 22 except for the last 20 bytes. However, the client receives from the security manager the capability + Cap_key which together constitute the credential. The Cap_key is what you call the 'digital signature' in Table 22 which should actually be 12 bytes instead of 20.

The OSD can compute the Cap_Key from the capability. Recall that the client sends, in addition to the request, the ReqMAC (a MAC on the request, computed with CAP_Key). Hence, by validating ReqMAC, the OSD ensures that Cap_Key is indeed correct.

IBM 32) Error in Model - Only Capability is sent to OSD in the CDB (Accepted, Substantive) [48]
page ??, clause ??

I looked at it again together with Michael. We still think that:

1. only bytes 0-87 of Table 22 are sent in the CDB. The last 20 bytes are the CAP_key which is *never* sent in the clear.

2. As a consequence, 3rd paragraph on page 41 is inaccurate as the server does not compare the MAC it recomputes against the last 20 bytes it receives.

5. Panasas

David Nagle from Panasas submitted the following comments.

Panasas 1) “Create Attributes Page” command and Proposed Attribute Templates (Unresolved) [49]

pages 79-80, 6.18

Discussion: Currently, the T10 Spec requires that each attribute page be explicitly created, using the Create Attribute Page command. This requires the higher-level software (e.g., file system) to issue multiple commands on object creation: 1) CREATE to create the object; 2) one or more CREATE ATTRIBUTES PAGE commands for each the object’s associated Attribute Pages. Multiple messages per object create is a significant performance problem for higher-level software using OSDs. Therefore, we recommend that all attribute pages and attributes implicitly exist once an object is created. Object creation (OSD CREATE with a SetAttr) allows a single command to both create an object and populate any attributes necessary. Additional attributes can be populated using either an explicit SET ATTRIBUTE command or via a SetAttr associated with another command.

In the original proposal from the Attributes Working Group, any attribute could be read or written at any time, and only standard-defined attributes were “predefined”. However, this level of flexibility imposes difficulties on efficient OSD implementations. Specifically, the OSD needs to locate commonly accessed attributes   Therefore, we recommend that the OSD provide a mechanism so that higher-level applications can disclose which attributes will be commonly accessed. We call this mechanism “Attribute Templates”, which are defined as follows:

- All attributes in the attribute space always exist
  - i. Any attribute can be written at any time
  - ii. Reading an undefined attribute will return a value of zero with zero length
- Commonly used attribute pages and their associated attributes can be “pre-defined” using a new entity called the “Attribute Template”
  - i. An Attribute Template can define multiple attribute pages and attributes
  - ii. There can exist multiple Attribute Templates
  - iii. Attribute Templates are created at root-object or partition-object creation time (see discussion below)
    1. The definition of the template is sent in the data-out phase of the FORMAT or CREATE OBJECT group command using a serialized representation of the template information using List entry serialized format (see table 98)
  - iv. Attribute Templates are stored in partition-objects (formally group-objects) for User-object Attribute Templates, or stored in the Root Object for Partition-object Attribute Templates
  - v. Attribute Templates have their own namespace
- Attribute templates may not be modified after they are created
- The OSD Create command can reference a root- or partition-object attribute template number when creating user-objects (likewise for the Create Group command)
- The Attribute Template page will define the attribute size for each attribute.
  - i. SetAttr of size greater than or less than the defined size will fail
  - ii. Any GetAttr will retrieve the entire attribute (all bytes defined by size) unless otherwise specified by the GetAttr
- Any attributes not defined by the Attribute Template still exists
  - i. GetAttr will return null with zero length
  - ii. SetAttr will create the attribute
- Attributes not defined by the Attribute Template are variable size, and that size can changed by a SetAttr command
  - i. The OSD is responsible for storing the length of each attribute. For attributes not defined by the Attribute Template, the attribute length is defined by the most recent SetAttr

Other comments: Since attribute reference pages are infrequently created and need to be checked on creation for valid references, the CREATE ATTRIBUTES PAGE command should be renamed the CREATE ATTRIBUTES REFERENCE PAGE command and used for creation of attribute reference pages.

Also, we suggested that Attribute Templates only be created at device or group creation time (CREATE OBJECT or FORMAT) to allow the OSD to optimize attribute space management across all objects within a device (or group).

Panasas 2) Attribute and Data Ordering (Accepted, Substantive) [50]
pages 15-16, 4.2.2 and 4.2.3
see also: comment Seagate 17)

Discussion: Data-in and Data-out buffer ordering is currently: 1) attribute list and attributes, followed by 2) data. This makes it difficult for the OSD to place the data into page-aligned buffers because the OSD must skip past the attributes before reading the data. To ease placement of data in page aligned buffers, data should immediately follow the SCSI CDB, with attribute lists and attributes placed the end of the Data-in and Data-out buffers.

For systems that move the entire Data-in/out buffer into RAM, this allows for immediate placement of page aligned data. Attribute lists and attributes appearing at the end of the buffer may not be page aligned. This, however, should not be a performance problem because attribute lists and attributes do not benefit from page alignment optimizations (e.g., page flipping, direct-data placement), because they are parsed by the CPU

Panasas 3) Atomicity of Set Attributes writes (Unresolved) [51]

pages 79-80, 6.18

Discussion: It is unclear from the spec what the maximum size of a write with SetAttr command can be while providing atomicity guarantees (i.e, the entire command either succeeds or fails). Clearly large writes (e.g., multi-megabyte) will find it difficult to guarantee atomicity. Therefore, the spec either needs to define a maximum size or provide some mechanism by with higher-level software can query the OSD for the vendor supported maximum size.

This topic also broaches the larger issue of error reporting and recovery, which are not well defined in the current standard. We propose that the immediate issue of atomicity of writes referred to the working group discussing recovery issues and a solution worked in the context of the broader problem.

Panasas 4) Attribute Directories (Unresolved) [52]

pages 85-86, 7.1.2.3 and 7.1.2.4

Summary: Individual object attribute directories should only contain information for attribute pages contained in the individual object.

Discussion: Currently, the Root and Group (i.e., partition) attribute directories contain entries for each Group and User object page. This was not agreed to in the Attributes Working Group. Moreover, the current definition means that two objects in the same OSD cannot use the same attribute page number for different information. Even if the higher-level software systems never intend on sharing information, once an application has claimed an attribute page within a group, all objects within the OSD must use the same attribute page definition.

We believe it is unnecessary and undesirable for Group and Root objects to hold directory entries for the attribute pages of lower-level objects. We believe that object directories should only contain references to attribute pages within the corresponding object.

Panasas 5) Size of Object Attribute Name Space (Unresolved) [53]

pages 19-22, 4.6.3.2

Discussion: The attribute name space is currently divided into 4 regions {256 defined by standard pages, 32K other standard defined pages, 32K defined by OSD mfg pages, 1G dynamic pages, and 1G vendor specific pages.

First, we do not understand the difference between the “defined by standard” and “defined by manufacturing product spec” pages. Second, we believe that the name space for dynamic pages is too large, and that most pages will be vendor specific not dynamic.

Finally, the semantics of dynamic pages is not well defined. We expect that most applications will want to share page numbers across all objects within a partition, within an OSD, or within a group of OSDs. This is not possible in the current design (and is really only possible with pre-assigned pages), which is one of the reasons why we believe that most systems will rely on pre-assigned pages. If a higher-level software system is assigned different dynamic pages (per object), then that system must either: 1) maintain an external catalog of {object, dynamic page} pairs or, 2) search through each object’s directory for the appropriate pages.

Moreover, there is nothing that guarantees that different higher-level software structures uniquely identify the dynamic pages they have created. Two different applications could use the same attribute directory name for a page, making it impossible for the application to know that it owns the corresponding attribute page.

We believe that dynamic pages and the attribute name space needs more discussion before being finalized in the spec.

Panasas 6) Large Capability and Where To Store It (Deferred to OSD-2) [54]

Global

Discussion: The SCSI CDB limits a command’s capability to 108 bytes. However, capabilities themselves can be much larger. To allow OSDs to support larger capabilities, the security working group originally proposed to cache large capabilities at the OSD, shipping the large capability in an OSD Session. In section 5.1.2.1.2.2, the T10 spec defines the method for referencing large capabilities … using a table of 2^128 entries. However, OSD Sessions are not well defined in general, and were originally designed for QOS guarantees, not the caching of capabilities.

We propose an alternative solution that fits within the current infrastructure; when employing a large capability, that capability should be stored on a dynamic attribute page. The SetAttr() command would send the large capability to the dynamic attribute page, and the OSD could return the page # and attribute # where the capability was stored. The capability could be stored on a partition-object or root-object, enabling it to be used by commands that reference lower-level objects (e.g., storing a large capability on a partition-object would allow any command that accesses objects within the corresponding partition to reference the large capability).

To secure this SetAttr command, the system could either: 1) use the large capability itself; or 2) use a small capability that granted SetAttr() permission to a dynamic attribute page.

Any future command that needed to reference the capability would use the dynamic attribute page #/attribute # to reference the capability, securing the command with a signature that could only be generated by a holder of the large capability.

Editor’s Note: If the statement made in comment IBM 26) is to be believed, the concerns described in this comment do not apply to this version of OSD.

Panasas 7) Get Attributes Parameters (Unresolved) [55]

page 40-41, 5.1.2.3 and 5.1.2.4

Discussion: Section 7.1.3.2 and 7.1.3.3 allows one to retrieve attributes from multiple objects. Touching multiple user objects requires accessing multiple distinct blocks on disk and potentially locking multiple objects. For commands that modify data, the failure semantics are not specified and troublesome. Currently there seems to be few usage cases for a single command touching multiple objects, therefore we recommend against enabling such operations.

We do recommend, however, that a single command be able to reference attributes on associated partition- and root-objects. There are many use cases, including Attribute Templates on OSD Creates, references dynamic-attributes that store large capabilities, modifying partition-object timestamps, and modifying partition- and root-object capacity-used attributes.

Further, Table 96 (the List entry format) should be modified to not permit specifying OBJECT_GROUP_ID and USER_OBJECT_ID for each attribute. Instead, since the OBJECT_GROUP_ID and USER_OBJECT_ID are implicit in the command, the List entry format only needs a bit vector to specify if the attribute to get or set is a user-, partition- or root object attribute.

Panasas 8) Get and Set Attribute Parameters (Accepted, Substantive) [56]

Page 40-41 & 43-44, 5.1.2.3, 5.1.2.4, 5.1.2.12, & 5.1.2.13
see also: comment Seagate 16)

Discussion: There are two types of GetAttr fields: 1) get 2 pages and 2) get one page and a list of attributes. Both consume 12 bytes in the CDB. The common case, however, will be either: 1) a single page or 2) a list of attribute/page pairs. Therefore, we recommend that we shrink the CDB fields from 12 bytes to 6 bytes and only encode a single page or list {length of list, length of buffer space}.

We should also do similar encoding on the Setattr fields (section 5.1.2.12), encoding either a single page or a list of attributes. This should reduce the CDB fields from 20 bytes down to about 12 bytes. (see notes written in document for more details)

Editor's Note: This comment will be resolved as described in comment OSD TWG 3).

Panasas 9) Security Attributes Page (Unresolved) [57]

Global

Discussion: There is no Security Attributes Page currently defined. The security page is necessary to store the object-version-number for each object, which is the standard method for invalidating a capability before its expire time. We ask that the Security Attribute Page be defined with an object-version-number attribute of size 8 bytes.

Panasas 10) Create command and the Create and Write command (Unresolved) [58]

pages 51-54, 6.4 and 6.5

Discussion: The CREATE command has a 28 byte REQUEST USER_OBJECT_ID field, which we believe is an error in size.

The Create And Write Command allows a user to simultaneously create an object and write data to the object. The command however has two problems. First, its capability is only 16 bytes in size, which is different from every other commands 108 byte credential. Second, the command only allows GetAttr's on pages and SetAttrs on values format. We do not believe it should have this limitation when other commands allow multiple GetAttr and SetAttr formats.

Panasas 11) Root object “Used Capacity” attribute (Unresolved) [59]

pages 87-88, 7.1.2.6

Summary: Change the root object’s “Used Capacity” attribute to be the sum of all capacity used within the device.

Discussion: The “Used Capacity” attribute only accounts for capacity used by the root object. However, in Group objects, the “Used Capacity” attribute sums the capacity used by all user-objects contained in the group-object. We recommend that the root object’s “Used Capacity” attribute sum all capacity used by user-objects, group-objects and the root object.

Panasas 12) Root object “OSD Security Level” attribute (Unresolved) [60]

page 87, 7.1.2.6
see also: comment IBM 28) and comment Seagate 29)

Discussion: The “OSD Security Level” attribute encodes the current level of security enforced by the OSD. Encoding a security level in the root object has two effects. First, all objects within a device share the same level of security. In the original proposal, capabilities encoded the minimum level of security, permitting different security levels for different objects. Since OSDs may be shared by numerous managers, it is possible that an OSD will need to provide different levels of security service. Further, the different levels of security may be required depending on the type of operation (e.g., changing root-object attributes vs. reading user-object data). Therefore, we believe we should not abandon encoded minimum security levels in each capability.

However, we do believe that device- or partition-objects should also be able to encode minimum levels of security. Therefore, we recommend that both device- and partition-object security pages include a minimum security level field, which could be set to NONE.

Editor’s Note: This comment is accepted in principle but the details of implementing its resolution need to be worked out. In general terms, the "minimum security level" concept will be applied. The Root object minimum security level will specify enforcement for accesses to the root object and provide a source for minimum security level inheritance to partitions. The new partition minimum security level will do for each partition what is currently defined for the root (plus the minimum enforcement concept).

Panasas 13) Group and User-object Information Attributes Page (Unresolved) [61]

pages 88-90, 7.1.2.7 and 7.1.2.8

Discussion: There is a clause at the end of each section, which states

> No page format is specified for the Group Information attributes page. If a CDB get or set attributes field specifies the page number of the Group Information attributes page, the command shall be terminated with a CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.

We do not understand this statement. It seems to imply that any set or get attributes command accesses the Information Page, the command will generate an error. We would like a clarification on this issue.

Panasas 14) Command with GetAttr and/or SetAttr and execution ordering (Unresolved) [62]

page ??, clause ??

The spec currently does not specify the ordering of when GetAttrs or SetAttrs occurs on commands with GetAttrs and/or SetAttrs. To clarify this issue, we suggest the following ordering: 1) Command and SetAttr occur at the same time, followed by 2) GetAttrs. This allows GetAttrs to retrieve updated information (e.g., capacity used) after the command has completed.

We also recommend that for standardized attributes, the standard may define that the attribute value before the command executes be returned by the GetAttr. This would only be for specific standard defined attributes, where the definition explicitly stated that an GetAttr would fetch an attributes value before command execution. Examples for this behavior are: 1) capacity used before write and 2) logical length before write.

Panasas 15) Size of object address space (Unresolved) [63]

page ??, clause ??

Discussion: The spec currently defines 2^128 objects within every partition-object (aka group), but an object can only store 2^64 bytes. We believe that we only need 2^64 objects w/in any partition.

Also, in Section 7.1.2.9, the Root resources attribute page specifies a "user objects per group count attribute", which is a 32-bit number. Isn't that number too small (limiting a group to only 4G objects? We would recommend allowing any group to have up to 2^64 objects. Further, we do not believe that the root-object should restrict the number of user-objects per partition. Likewise, in section 7.1.2.10, the Group Resources Attributes Page has an "Object count" field of only 4 bytes. We would recommend an 8-byte integer.

Panasas 16) Last-Access time (Unresolved) [64]

page ??, clause ??

Some standard-defined, OSD-managed attributes should for command control of updates. For example, the Time Stamp attributes page ATTRIBUTE_ACCESSED_TIME and DATA_ACCESSED_TIME attributes will currently be updated on every access. This can be a burden to the OSD and significantly impact higher-level software performance. We recommend that attribute update behavior be controllable, allowing higher-level software to turn off the default (i.e., update) behavior to enhance performance. In general, this will benefit the performance of any command that does not modify data but does modify attribute(s).

Panasas 17) Group Resources Attribute "Remaining Capacity" (Unresolved) [65]

page 92-93, 7.1.2.10

The "Remaining Capacity" attribute is currently defined to be the minimum of: 1) capacity quota - used capacity or 2) total capacity in root attributes page. We believe this is unnecessary and imposes a burden on the OSD. Instead, we believe that every user-, partition- and root-object should specify the capacity quota and used capacity. From these two values, the higher-level software can then compute any other information necessary.

Panasas 18) Terminology (Unresolved) [66]

pages 5-8, 3.1 & 3.2
see also: comment IBM 1)

Discussion: There appears to be numerous usages for the terminology used to describe capabilities and credentials. The Terminology document defines them as:

Capability (Cap) - a data structure describing the operations a principal may do on an object and including an object selector
Credential - a capability and its cryptographic seal of authenticity issued to a client by a security manager (includes also some randomization elements - nonces)
Capability arguments (CapArgs) - the Capability, the Object selector and the nonces
Capability key (CapKey) - the MAC on the capability arguments
RequestMAC - the MAC on the command sent to an object store
ManagerNonce - a nonce assigned by the security manager (e.e., in capability arguments)
ClientNonce -a nonce assigned by the client (e.g.,in request arguments)

In contrast, the T10 spec uses the term “REQUEST_DIGITAL_SIGNATURE” to refer to the cryptographic seal on an client-to-OSD request. The term “CREDENTIAL_DIGITAL_SIGNATURE” refers to the cryptographic seal on a credential. The T10 “Credential” includes the CREDENTIAL_DIGITAL_SIGNATURE, but not the request nonce nor the REQUEST_DIGITAL_SIGNATURE. Further, the CREDENTIAL_DIGITAL_SIGNATURE is currently encoded in every command (in the Credential Format Table 22). However, the CREDENTIAL_DIGITAL_SIGNATURE should only be sent from the manager to the client when a capability is requested.

We suggest some clarifications based on what messages are produced and used. First, when a security manager grants permission to a client (to access an object or set of objects), the security manager passes a **Capability + Capability Key** to the client. The **Capability** includes all information necessary for the client or OSD to determine which operations a client has been granted access, including the capability-nonce. However, the Capability does not include the Capability Key. The Capability Key is a separate entity, that is only passed from the security manager to the client, whereas the Capability can be passed from the client to the OSD.

The **Credential**, is the **Capability + Nonce(s) + Digital Signature** over the CAP+Nonce(s), which is passed from the client to OSD, allowing the OSD to verify that permissions.

This discussion somewhat follows the definitions given in the Terminology document, but there is also overlap in those definitions. Therefore, we propose the following:

Capability (CAP) - a data structure describing the operations a principal may perform. The capability may be passed from security manager to client, and from client to OSD in the clear. The fields of the Capability include:

Object-group ID
User-object Descriptor
User-object ID
Length
Starting Byte Address
User-object Creation Time
Permission Bit Mask
Expire Time
Object Version # (aka sec. window verbs)
Key Version
Boot Epoch
Protocol Version
Hash Algorithm Used
Minimum Security
Audit Field

Capability Key (CAPKEY) - The MAC over the entire Capability. Send over a security channel from the security manager to the client.
Credential - CAP + CAPKEY
RequestMAC - the MAC on the command sent to an object store
ManagerNonce - a nonce assigned by the security manager (e.e., in capability arguments)
ClientNonce -a nonce assigned by the client (e.g.,in request arguments)

Note: Should we include the object-version-# in table 27, the user object descriptor?

Panasas 19) Credential Format (Accepted, Substantive) [67]
page 35, 5.1.2.1.1, table 20
see also: comment IBM 31) & comment IBM 32)

Note: References in original comment are from OSD r07 not OSD r07a.

The Credential Format (Table 22) includes a Credential Digital Signature Field (20 bytes). The Credential is the set of bits shipped in every CDB to the OSD. However, the Credential Digital Signature is the signature (i.e., CAPKEY) sent from the Security Manager to the client (over a secure channel), and which the client uses as its key for signing the Credential (see Table 21, Request Digital Signature). Therefore, we ask that the Credential Digital Signature be removed from the Credential.

Note: This is an example of where terminology has caused some confusion. From the definition above, the Credential is the capability+seal sent from the Security Manager to the Client. Therefore, the Credential format is technically correct. However, the Credential Format cannot be used in the CDB because sending the Credential Digital Signature discloses the key used to sign the command. The Credential Format is what the Security Manager would send to the client over a secure channel (e.g., Credential). Therefore, we recommend that the CDB include a Capability Field and a Request Digital Signature (i.e., RequestMAC).

Editor’s Note: This comment will be resolved as described in comment IBM 31) and comment IBM 32).

Panasas 20) Fields important to include in the Capability (Unresolved) [68]

page 35, 5.1.2.1.1

Discussion: The current capability does not include several pieces of information that are important to support security and reliability/recovery. These include:

Boot epoch

A 32-bit unsigned int, maintained by the OSD that uniquely identifies each epoch that the OSD is operational. In practice we expect that this will usually be implemented as a counter, maintained on stable media, that is incremented at every boot. The boot_epoch field in the capability must either be zero (A reserved value) or must exactly match the boot epoch value in the OSD. The OSD's current boot epoch is accessible using the boot epoch attribute in the security page on the root object.

Protocol Version

A 32-bit unsigned int identifying the protocol revision to which this capability applies.

Hash Algorithm Used

16-bit unsigned int that identifies the hash algorithm used to sign the credential. The value used must appear in both the device's Hash algorithms supported attribute and its Hash algorithms allowed attribute.

Minimum Security

16-bit bit-mask that identifies the level of security required to be employed in transactions to which this capability applies. This is a combination of the following values:

Privacy capability
Integrity args
Privacy Args
Integrity Data-in
Privacy Data-in
Integrity Data-out
Privacy Data-out
Integrity response
Privacy response

Audit Field

Used for Nonce (see IBM and Gibson email)

The spec has a field called the SECURITY_WINDOW, which we believe was originally called the capability version.

- Should not require the system to always increment by one (should be setting by the outside world)

The current spec includes a CREDENTIAL_CREATE_TIME field. We do not know what purpose this field serves.

An alternative to the Hash Algorithm Used field, is to set the Hash Algorithm used on a partition (via a new partition-object hash-algorithm-used attribute). This allows the higher level software to select a hash algorithm (from those supported in the OSD), but forces that algorithm to be applied to all commands that access objects within the partition-object.

Panasas 21) Capabilities Permission Bits (Unresolved) [69]

page 37, 5.1.2.1.1, table 23

SetAttr's (either as the setattr command or command+setattr) can set attributes on user-object, partition-object, and root-object attributes. The capability should be able to encode which objects (user-, partition, and root-object), a command may set or get. This granularity of access needs to be enumerated in the Capabilities Permission Bits to include {setattr user-object, getattr user-object, setattr partition-object, getattr partition-object, setattr root-object, getattr root-object}.

We believe that the well-known object (WK_OBJ) (see Table 20 and accompanying text) does support some of this functionality. However, we recommend that Table 23 be expanded to explicitly enable permission bits that apply to user-, partition-, and root-objects.

Panasas 22) Key Version field too large (Unresolved) [70]

page 35, 5.1.2.1.1, Table 20

The key version field is currently 32-bits long. We do not understand the need for this field to be larger than 16-bits.

Panasas 23) Timestamp bypass (Unresolved) [71]

page ??, clause ??

Some timestamp attributes (e.g., last time accessed, last time modified) are currently set automatically by the OSD. This can be very costly to the performance of operations that do not modify data (e.g., read), because it forces the OSD to perform a write. Further, many file systems do not require all timestamps to be up-to-date, allowing some to slip. For example, a file system reading a large file may only care that the last read modify the last-access time.

We propose that the OSD support a Timestamp bypass mechanism that allows the system to turn off timestamp updates. Encoded in the CDB, the timestamp bypass field would instruct the OSD to not update the last-accessed time on an object. To avoid security problems, the Capability for the command would have to grant permission to perform a timestamp bypass.

If supporting timestamp bypass in the CDB is too difficult for the OSD, then we propose that the LAST_ACCESS_TIME timestamp updates be optional, set by a flag on the partition-object and applied to all objects in the partition.

Panasas 24) Credential Creation Time (Unresolved) [72]

page 35, 5.1.2.1.1

We do not understand the need to encode the creation time of a credential and suggest that it be removed.

Panasas 25) Credential Nonce (Unresolved) [73]

page 35, 5.1.2.1.1, Table 20

There are currently several different types of nonces used in the system. We recommend that the nonce encoded in the Credential (ManagerNonce) be called something else and located in the Credential’s audit field.

6. Seagate Technology

Sami Iren from Seagate Technology submitted the following comments.

Seagate 1) Attribute inheritance wording is unclear (Unresolved) [74]

Page 17, 4.6.2.1

I would rephrase the following sentence:

> ”The default attributes for a group object are inherited from the attributes in the root object”

as

> “The default values for some of the group object attributes are copied (inherited) from the corresponding attributes in the root object as specified in this standard”.

Same thing for user object attributes.

Seagate 2) Last attribute is FFFF FFFEh (Unresolved) [75]

Page 21, 4.6.3.3, p 1

The attribute number range is incorrect. The high end is FFFF FFFEh. By definition, FFFF FFFFh is used to refer to all attributes (the second paragraph states this correctly).

Seagate 3) Attribute directories are irrelevant (Unresolved) [76]

Page 21, 4.6.3.4

Attribute directories are irrelevant with the new “all attributes exist all the time approach” (see the OSD attributes Draft v 0.2)

Seagate 4) Why limit sessions to READ, WRITE, and APPEND? (Unresolved) [77]

Page 22, 4.6.4.2

Why are we limiting the use of sessions to READ,WRITE, and APPEND? I think we should say “any command that accesses data”.

How would the CLOSE command specify that all non-default sessions should be closed, and why would this be allowed?

Seagate 5) 'a' s/b 'an' (Accepted, Editorial) [78]

Page 22, 4.6.4.2, 2nd to last p

should read “Once an object session …” (“an” NOT “a”)

Seagate 6) Terminology inconsistent between Figure 4 and text (Unresolved) [79]

Page 23, 4.6.5.1, Figure 4

Try to be consistent with the text. Text uses the term “device server”, figure uses OSD Device. Text uses the term “service delivery subsystem”, figure does not explicitly shows it.

Seagate 7) Remove 'that' & insert 'is' (Accepted, Editorial) [80]
Page 23, 4.6.5.1, p 5

The sentence should read: "The device server validates each command received from an application client to confirm that:" (drop the first that after "validates"). In the same paragraph, option b, the word "is" is missing right after "the command key that …. ".

Seagate 8) Too many 'may's (Accepted, Editorial) [81]
Page 24, 4.6.5.2.1, p 1, s 1
same problem identified in comment IBM 9)

The first sentence should read: "An OSD request maybe made …" (drop the first "may" after OSD)

Seagate 9) Security Level 1 1st sentence is nonsense (Accepted, Editorial) [82]
Page 24, 4.6.5.2.2, p 2

First sentence of the paragraph does not make sense.

Editor's Note: Comment IBM 7) requests that the definitions for levels 1 and 2 be recombined so that they match the Security Document (see Comment OSD TWG 2). The problem noted in this comment will be resolved as part of the rewrite needed to resolve comment IBM 7).

Seagate 10) Level 4 overview sentence does not make sense (Unresolved) [83]

Page 25, 4.6.5.2.5, p 4

The sentence starting with "Level 4 provides for …"): sentence does not make sense.

Editor's Note: The sentence in question currently reads.

> Level 4 provides for the application of digital signatures to every datum exchanged between the application client and OSD.

The problem is unclear to me. Perhaps 'the application of' should be replaced by 'applying'. Alternatively, perhaps 'datum' should be replaced by 'byte'.

Seagate 11) Insert cross reference to table 15 (Accepted, Editorial) [84]
page 27, 4.7.1

Change "the commands in to initialize" to "the commands in Table 15 to initialize"

Editor's Note: Accepted as written, except that 'table' will not be capitalized.

Seagate 12) Change 'device server' to 'OSD device' (Rejected) [85]
page 27, 4.7.1

Change "The device server shall accept OSD mandatory" to "The OSD device shall accept OSD mandatory"

Reason for Rejection: In the SCSI model, device servers are the peers of application clients and are the entities that accept and process commands.

Seagate 13) Keep subclause 4.7.2 (Discovery and Configuration) (Rejected) [86]
Page 28, 4.7.2

This section needs to be in the specification. If the OSD device is to identify itself to all initiators, then the commands to be used, timing, ways to identify all initiators, etc must be defined.

Reason for Rejection: See comment Editor 2) for a discussion of why this comment is being rejected.

Seagate 14) Example needed in the OSD model (Unresolved) [87]

Pages 28-30, 4.7.4

This is a very needed section but should be more detailed.

Editor's Note: There are those in T10 who will argue that this example should be removed. Perhaps it can be retained based on the newness of OSD. However, someone must assume responsibility for updating it.

If the necessary updates are not provided in time for the T10 Letter Ballot review, 4.7.4 will be removed in its entirety.

Seagate 15) Linked Command Support (No Action) [88]
Page 33, 4.8, 6th paragraph

Are we supporting linked commands?

Editor's Note: SAM-3 says that linked command support is optional. However, the reservations specification needs to describe the interaction between linked commands and reservations.

Seagate 16) Get/Set Parameters Changes (Accepted, Substantive) [89]
Page 40-41 & 43-44, 5.1.2.3, 5.1.2.4, 5.1.2.12, & 5.1.2.13
see also: comment Panasas 8)

Get/set attributes parameters should get/set either a single page or a list, not two pages or a page and a list.

Editor's Note: This comment will be resolved as described in comment OSD TWG 3).

Seagate 17) Data first in Data-In/Out Buffers (Accepted, Substantive) [90]
pages 15-16, 4.2.2 and 4.2.3
see also: comment Panasas 2)

Flip the order of data and attributes in Data in/out buffers. Data should come first.

Seagate 18) Last Byte number wrong in CREATE AND WRITE format (Accepted, Editorial) [91]
page 53, 6.5, table 44

The last row should read 235, not 145.

Seagate 19) Remove the CREATE ATTRIBUTES PAGE command (Unresolved) [92]

pages 55-56, 6.6

The CREATES ATTRIBUTES PAGE command should be removed. We are assuming all the attributes always exist and they do not have to be explicitly created. For the commonly accessed attributes a new concept called "attribute templates" is proposed. The following commands should be added for this new concept:

- CREATE ATTRIBUTE TEMPLATE
- REMOVE ATTRIBUTE TEMPLATE
- LIST ATTRIBUTE TEMPLATE

Seagate 20) Define FLUSH PARTITION & FLUSH OSD (Deferred to OSD-2) [93]

Global

We suggest defining that flushing a group (partition) object affects all user objects in that group (i.e., synching a partition) and flushing the root object is effectively a "sync" command on the whole device.

Editor's Note: Attempting to obtain the necessary consensus on this is inconsistent with an expeditious transition to T10 Letter Ballot for OSD.

Seagate 21) IMPORT USER OBJECT CDB needs more fields (Unresolved) [94]

pages 64-65, 6.11

The IMPORT USER OBJECT command does not allow user-specified object id on the destination OSD. Also the command is missing the credentials to be passed along to the source OSD.

Editor's Note: Resolving this comment as written is not workable. There are not enough bytes in a CDB to contain two credentials.

Seagate 22) Increase Index field size in LIST CDB (Accepted, Substantive) [95]

pages 66-67, 6.12

The index to the LIST command should be the unique object ids rather than the position of the ids within the sort order. Also, the index field should be 8 bytes rather than 4 BITS. 4 bits will not get us anywhere even with the current spec.

Editor's Note: This comment will be resolved as described in the resolution to comment Editor 4).

Seagate 23) Add Session Template to OPEN (Unresolved) [96]

pages 69-70, 6.13

The CDB should include a "session template id" which includes the default attributes/values of the newly created session. The way it is defined now, clients have to do set attribute for all the object sessions although many of them might have the same characteristics. The following commands should be defined for the session templates:

- CREATE SESSION TEMPLATE
- REMOVE SESSION TEMPLATE
- LIST SESSION TEMPLATE

Seagate 24) Reduce WRITE byte count to 32 bits (Unresolved) [97]

page 81, 6.19, table 54

64-bit length field is too big for WRITE. It should be 32 bits.

Seagate 25) Increase Object Logical Length to 96 bits (Unresolved) [98]

pages 93-95, 7.1.2.11

64 bits is not enough for object size. Maybe make it 96 bits?

Seagate 26) Limit access available to list format attributes (Unresolved) [99]

Page 91, 7.1.1, 2nd to last p, last s

Replace this sentence with, "Using the list format, any attribute associated with the user object specified by a service action, the object group (partition) of which that user object is a member, and the root object is accessible."

Giving access to attributes of other objects poses security problems.

Editor's Note: The current wording allows access to any attribute known to the OSD.

Seagate 27) Remove directory pages and definitions (Unresolved) [100]

pages 85-86, 7.1.2.3, 7.1.2.4, & 7.1.2.3

Root directory, Group directory, and User object directory pages are NOT needed. All the attribute pages exist all the time and there is no need to keep these directories.

Seagate 28) Root 'Used Capacity' attribute should count all user objects (Unresolved) [101]

page 87, 7.1.2.6, 2nd p from bottom of pg

The [root object] used capacity attribute should contain the number of bytes used by all the objects on the device including the user objects. Currently it only keeps the bytes used by the root object.

Seagate 29) Add Security Level to Group attributes (Accepted, Substantive) [102]

page 88-89, 7.1.2.7
see also: comment IBM 28)

Group information attributes page should have a new attribute called "OSD security level" just like the root information attributes page does. This will determine the security level of the group (partition).

Editor's Note: This comment will be resolved as described in the response to comment Panasas 12).

Seagate 30) Wrong attribute number for Group Username (Accepted, Editorial) [103]

page 89, 7.1.2.7

The attribute number for username is inconsistent with the text. Table says Ch, text says 10h.

Editor's Note: The table is right, the text will be changed.

Seagate 31) Wrong attribute number for User Object Username (Accepted, Editorial) [104]

page 89, 7.1.2.8

The attribute number for username is inconsistent with the text. Table says Ch, text says 10h.

Editor's Note: The table is right, the text will be changed.

Seagate 32) Does User Object Used Capacity report space actually used? (Unresolved) [105]

page 89, 7.1.2.8

How does the Used Capacity attribute deal with sparse objects? Does it report the actual bytes used on the disk, or the logical size of the file? Since we are talking about capacity and quotas, it should the latter. In this case, it would be helpful to have another attribute that reports the actual space used.

Seagate 33) Increase Group Count size from 4 to 8 bytes (Unresolved) [106]

pages 90-91, 7.1.2.9

The Group count field should be 8 bytes, not 4.

Seagate 34) ’total capacity' s/b 'remaining capacity' (Accepted, Substantive) [107]

page 92, 7.1.2.10, list entry b)
see also comment Seagate 36)

The list entry should read: "The value in the available capacity attribute in the Root Resources attributes page." (available capacity, NOT total capacity).

Editor’s Note: Actually, there is no attribute called 'available capacity'. However, there is an attribute called 'remaining capacity'.

Seagate 35) Move 'Object Logical Length' to User Object Info Page (Unresolved) [108]

pages 89-90, 7.1.2.8 & pages 93-95, 7.1.2.11

Shouldn’t "Object logical length" attribute belong to Table 67 [User Object Information attributes page]? This is an information attribute rather than a resource attribute, right?

Seagate 36) Mention of Root Remaining Capacity is redundant (Rejected) [109]

page 94, 7.1.2.11, list entry c)

List item c should be removed. This is not necessary. Item b already checks for the root resources remaining capacity.

Editor’s Note: This is not an implementation definition. It is a statement of required actions. Providing a complete list increases the clarity of the requirements statement, even if it is redundant from an implementation point of view.

Seagate 37) REMOVE OBJECT GROUP should update Root Modification time (Accepted, Substantive) [110]

page 96, 7.1.2.12, p 6 on pg
see also: comment Editor 5)

Shouldn’t the data modified time be updated when a group is removed, too?

Seagate 38) Why return session id in the Current Command attributes page? (No Action) [111]
pages 101-102, 7.1.2.15

This is an unusual attribute page. Why do we need this (to return the session id?)? When we say "current command" are we making any assumptions on the number of commands we can execute in parallel?

Editor's Note: When a SCSI command completes with GOOD status, only one byte of "response" information is permitted and that byte is the one containing the GOOD status. The Current Command attributes page was created in order to return other information.

Using the cited Session Id attribute as an example, the OPEN command needs to return the OSD device server assigned session id. This is accomplished via the Session Id attribute and its reference in the Session attributes page (see 7.1.2.19).

Seagate 39) Update Attributes Lists definitions for restricted access (Accepted, Substantive) [112]
pages 110-115, 7.1.3 Global

All the tables on these pages should be updated to reflect the fact that we are not allowing access to attributes of other user objects. Only the current user object, its group (partition) object, and root object attributes are accessible.

Editor's Note: The entire 7.1.3 subclause will be reviewed and updated to reflect the new limitations on accessibility of attributes belonging to one object from another object.