ENDL
TEXAS

Date: 1 September 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.

r1 Added editor comments Editor 8) which renames 4.2 to reduce confusion, Editor 9) which removes all discussion of OSD policies, Editor 10) which reminds all that the requirements in the T10 Style Guide will be applied in r08. *Rejected*: IBM 2), Panasas 20), Panasas 25), Seagate 3), Seagate 24), and Seagate 27). *Deferred to OSD-2*: Panasas 1) and Seagate 23). *Resolved*: EMC 1), EMC 4), EMC 5), IBM 1), IBM 4), IBM 5), IBM 6), Panasas 3), Panasas 5), Panasas 7), Panasas 9), Panasas 11), Panasas 12), Panasas 13), Panasas 14), Panasas 15), Panasas 18), Panasas 22), Panasas 24), Seagate 1), Seagate 2), Seagate 6), Seagate 10), Seagate 14), Seagate 21), Seagate 33), and Seagate 26). Noted that comments Panasas 16) and Panasas 23) request the same change. Noted relationship between comments Seagate 25), Seagate 32), and Seagate 35). Updated the resolutions for all TWG working group provided documents to further specify how the documents will be incorporated. The updates in r1 are not final and future revisions of this document will modify them further.

r2 Completed the description of how SNIA OSD TWG documents will be incorporated. *Added*: Editor 11) and Editor 12). *Resolved*: IBM 24), Panasas 21), Seagate 4), and Seagate 19). *Deferred to OSD-2:* Panasas 16) and Panasas 23). Comment Seagate 25) was withdrawn by the author.

r3 Added complete results of issues put to vote. *Deferred to OSD-2:* Panasas 4). *Accepted:* Panasas 17) and Seagate 35). Glossary changes in IBM 1) updated. Added a requirement to remove all Editors Notes, Editor 13), note if this results in substantive changes it will be documented in a future revision of this document. Updated OSD TWG 3) to reflect comments from Sami Iren. Updated Panasas 12) to reflect comments from Sami Iren and issues encountered during the preliminary incorporation of OSD TWG 2). Updated OSD TWG 2) to add a request for a NONCE TIMESTAMP OUT OF RANGE additional sense code. Updated Panasas 2) to specify a buffer ordering that facilitates implementation of OSD on SCSI transport protocols that do not support modify data pointers.

Changes marked by change bars.

18484 Preston Road, Suite 102 PMB 178, Dallas, TX 75252 214-912-1373 972-596-2775 Fx ENDL_TX@computer.org

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

Rejected Comments List

No Action Requested, No Action Taken Comments List

Comments With Implementation Deferred to OSD-2

Substantive Comments Accepted As Proposed

Substantive Comments Accepted With Noted Changes

Accepted As Proposed Non-Substantive Comments List

Accepted With Noted Changes Non-Substantive Comments List

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) The Abstract, Acknowledgements, Log of Changes, Section 2 (Requirements), and Section 3 (Architectural assumptions) will not been incorporated because they contain explanatory material that is not appropriate for inclusion in a T10 standard.
2) Sections 4 (Proposed structure for ObSID), 5 (Proposed structure for PtID), and 6 (Proposed structure for OID) all mention registration of something with T10. T10 registers precisely one thing, Vendor ID. Except for specifying the use of specific values in the OSD standard, T10 will not 'register' anything else. No discussion of T10 registering things will be incorporated.
3) Section 4 (Proposed structure for ObSID) will be incorporated as follows:
   a) Based on the following statement in section 7 (Use of identifiers in object creation, deletion and access) "ObS selects a specific device (volume or LU in SCSI lingo) with the ObS-Selector. It implicitly selects the ObSID.", ObSID will be incorporated as a new attribute in Root Information attributes page,
   b) To support the concept of 'right extended with zeros', the glossary entry for zero-padded will be copied from SPC-3 with the cross reference changed to point to SPC-3. Then the phrasing 'right extended with zeros' will be replaced with zero-padded,
   c) The SPC-3 definitions for EUI-64 and NAA format identifiers will be copied to the definition ObSID and the standardization references they use will be copied to the glossary, references, and bibliography clauses or annexes as appropriate, and
   d) The name ObSID will be changed to something that will pass a T10 Letter Ballot, probably OSD System ID.
4) Section 5 (Proposed structure for PtID) will be incorporated as follows:
   a) The Group_Object_ID (now Partition_Object_ID) will be increased from 4 to 8 bytes. **Note:** this change affects tables 6 (Object_Group_ID value assignments) & 7 (Object_Group_ID and User_Object_ID value assignments), all CDB formats, the Group Information attributes page, and the User Object Information attributes page,
   b) The upper limit of the range of reserved Partition_ObjectIDs will be increased from Fh to FFFFh. **Note:** this change affects tables 6 (Object_Group_ID value assignments) & 7 (Object_Group_ID and User_Object_ID value assignments).
   c) The name of the Root Object will not be changed to WKPtID.
5) Section 6 (Proposed structure for OID) will be incorporated by increasing the upper limit of the range of reserved User_Object_IDs from Fh to FFFFh. **Note:** this change affects table 7 (Object_Group_ID and User_Object_ID value assignments). The name of User_Object_IDs will not be changed to OID.

6) The section 7 (Use of identifiers in object creation, deletion and access) hanging paragraphs, section 7.1 (Creating objects with client supplied IDs), and section 7.2 (Creating objects with ObS assigned IDs) will be incorporated as follows:
   a) Two of the reserved bytes in the CREATE CDB recovered in response to comment Panasas 10) will be used to define a NUMBER OF OBJECTS field with the semantics described;
   b) Specific wording will be added to require the requested USER_OBJECT_ID field to be zero if the NUMBER OF OBJECTS field is greater than one;
   c) Specific wording will be added to require that when more than one object is created by a CREATE command the objects shall have User_Object_IDs that differ from each other by one and that the lowest valued User_Object_IDs for any of the created user objects shall be the User_Object_ID placed in User_Object_ID attribute of the User Object Information page,
   d) The specification for the OBJECT_GROUP_ID field in both the CREATE and the CREATE AND WRITE commands will have text added requiring that the identified group (now Partition) be defined,
   e) Wording will be added to the description of the NUMBER OF OBJECTS field requiring immediate invalidation of the credential used to perform the CREATE operation after the operation is completed, and
   f) No other changes will be made because all other text presents information that is already covered in OSD r07a.
7) Section 7.3 (Creating/Deleting Partitions), section 7.4 (Object deletion), and section 7.5 (Object Access) will not be incorporated because they contain no information that is not already present in OSD r07a.
8) Section 8 (Temporary vs. Permanent objects) will not be incorporated. The section 9.2 of the OSD Grouping and Attributes document (see comment OSD TWG 3) defers the 'committed' attribute and garbage collection to the next OSD version, so that president will be followed here.
9) Section 9 (Additional security considerations) will not be incorporated since all security issues should be covered in the Object Store Security Document (see comment OSD TWG 2). However, it is noted that the requirement to multiple object creations to a single use of a capability is in response to a security issue.
10) None of Section 10 [Future work on Versioning (not for consideration for the current of this document)] will be incorporated.

Document 03-278r1 has been uploaded to further clarify the incorporation of 03-278r0. The two documents are identical except that 03-278r1 contains PDF markups that will guide the incorporation of 03-278r0.

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 0 (Revision History), 9 (References), and 10 (Appendix) will not been incorporated because they contain explanatory material that is not appropriate for inclusion in a T10 standard.
2) The Section 1 (Introduction) hanging paragraphs will not will not been incorporated because they contain explanatory material that is not appropriate for inclusion in a T10 standard.
3) Sections 1.1.1 (Basic Security Model), 1.1.2 (Trust Assumptions), and 1.3 (Requirements Summary) will not be incorporated because they were incorporated in OSD r07a. Errors during incorporation of this material are noted and resolved in several comments (e.g., comment IBM 7).

4) Sections 1.1.3 (Security Flow and Channel Requirements), 1.1.4 (Layered Approach to Protocol Definition), 1.2.5 (Privacy), 1.2.6 (Summary of Security Levels), and 1.4 (Limitations in the Proposed Version of Object Store Protocol) will not be incorporated because that contain material that is not appropriate for inclusion in a T10 standard.

5) The Section 1.2 (Levels of Security) hanging paragraphs, Section 1.2.1 (No Security), Section 1.2.2 (Level 1 – Integrity of Capability), Section 1.2.3 (Level 2 – Integrity of Command and Arguments), and Section 1.2.4 (Level 3 – Integrity of Data) will not be incorporated because they were incorporated in OSD r07a. It should be noted that these sections now contain statements about support at each level being optional. Such statements need not be incorporated in OSD r08 because an behavior that is not required is (by definition) optional.

6) Sections 2.1 (Introduction) and 2.2 (Cryptographic Building Blocks) will not be incorporated because that contain material that is not appropriate for inclusion in a T10 standard.

7) Sections 2.7 (Credential Invalidation), 7 (Security Manger – OSD protocol), and 8 (Key Management) will be used to create two new subclauses:

   • 4.5.6.x Interactions between the security manager and OSD (after the last subclause in 4.5.6)
   • 6.x SET KEY

   as follows:

   a) The hanging paragraphs in Sections 7 and 8 will provide introductory concepts for the new subclause;

   b) Section 9 of the OSD Grouping and Attributes document (03-280r1) states that the version attribute is not being defined in OSD r08. Therefore, the sections 2.7 and 7.1 discussions of invalidating Credentials by changing the Version Tag will not be incorporated. This leaves invalidating Credentials by key changes (see section 2.7) as the only Credential invalidation mechanism;

   c) The functional requirements in Section 7.2 (Clocks and Expiry Time) will be incorporated in 4.5.6.x;

   d) Section 8.1 (Requirements) will not be incorporated because it contains material that is inappropriate for a T10 standard;

   e) The information in section 8.2 (Key Hierarchy) will be incorporated in 4.5.6.x;

   f) Sections 8.3 (Key Exchange Protocol) and 8.4 (Using the standard protocol to Set Keys) will be used to construct the SET KEY command;

   g) Section 8.5 (Drive Initialization) will not be incorporated because it does not refer to use of the SET KEY command and thus must be outside the scope of the standard;

   h) The information in sections 8.6 (Storing Long Lived Keys) and 8.7 (Secure Computation) will be incorporated in 4.5.6.x;

   i) Section 8.8 (Parameterizing Cryptographic Primitives) will not be directly incorporated in the OSD r08 for the following reasons:

      • The presence of the integrity check value algorithm identifier in the CAP_Args is covered in the Capability definition below;
      • Since there is only one integrity check algorithm at this time, there is no need for an attribute defining the preferred usage of integrity check algorithms; and
      • The remaining text does not appear to be appropriate for a T10 standard.

   j) Text in sections 2.7, 7, and 8 that is not appropriate to a T10 standard will be reworded or not included.

   k) Section 2.3 (Key Management Overview) contains information that is either inappropriate for a T10 standard or repeated in Section 8. Section 2.3 will be ignored during the preparation of OSD r08.

8) Sections 2.4 (Capability Argument and Capability Key) and 2.5 (Anonymous Object Creation) will be used to:

   a) Redefine the structure and content of a Capability in 5.1.2.1. The fields in the Capability (as it will appear in each CDB) are:
      - Credential format
      - Integrity check value algorithm
        (Note: the above two appear in the Capability because they occupy only one byte, because there would appear to be no other way for the device server to the integrity check value algorithm, unless the device server also has access to all credentials distributed by the Security Manager, a message flow not provided for in section 2.6, and because section 8.8 lines 1212 & 1213 indicate that integrity check value algorithm is part of CAP_Args)
      - Rights string (Type, Permissions Bit Mask, User Object ID)
        see also comment IBM 26)
      - Key version
      - Nonce (Note: this nonce is different from the security nonce described in 4.6.5.4)
      - Expiration time

   b) Insure that the definition of Key version provides for a CHECK CONDITION if the capability key version fails to match the key version being used by the device server,

   c) Create a new subclause (inserted after 4.6.5.3) defining the structure and content of a Credential **Note:** Section 9 of the OSD Grouping and Attributes document (03-280r1) states that the version attribute is not being defined in OSD r08. Therefore, 32 bits will be reserved in the credential format for a Version Tag but their usage will not be defined. The fields in the Credential (including those called Capability fields that later 2.4 paragraphs describe is not being part of the formal Capability in the CDB) are:

      - OSD System ID
      - Partition ID
      - Creation time
      - Reserved (Version tag)
      - Credential integrity check value

   d) Create a new subclause following the one just described in which the process by which a device server validates a received credential is specified and add a reference to the new subclause in 5.1.2.1 (Capability format),

   e) Update the Clock attribute in the Root Information attributes page (7.1.2.6) to match the format defined for Expiration time, also

   f) Text in section 2.4 that is not appropriate to a T10 standard will be reworded or not included.

9) Section 2.6 (Message Flow) will not be incorporated because it was incorporated in OSD r07a. The resolutions for various other comments (e.g., comment IBM 31) will be applied to correct errors made in OSD r07a.

10) With three exceptions, Section 2.8 (Security Related Error Status) will not be incorporated because the listed error status codes are inappropriate or redundant in the context of SCSI sense data. The nature of the problems are as follows:

    - NOT_SUPPORTED_CREDENTIAL_TYPE – No OSD defined command includes credential type as a field.
    - CAPABILITY_MISMATCH, INVALID_MAC, INVALID_VERSION, INVALID_KEY, EXPIRED_ CREDENTIAL, INVALID_NONCE, and NONCE_NOT_UNIQUE – This information can be determined from the SENSE KEY SPECIFIC field for an ILLEGAL REQUEST sense key.
    - INSUFFICIENT_RESOURCES – The SYSTEM RESOURCE FAILURE and INSUFFICIENT RESOURCES are already defined for this purpose.

- INVALID_MESSAGE_STRUCTURE – The ILLEGAL REQUEST sense key is defined for this purpose as are the INVALID FIELD IN CDB and INVALID FIELD IN PARAMETER LIST additional sense codes and the SENSE KEY SPECIFIC field.

Two of the exceptions involve conditions that are not strictly errors in CDB fields and conditions that require more significant recovery actions. They are:

- CAPABILITY_BLOCKED – Request ASC/ASCQ 24h/04h for SECURITY AUDIT VALUE FROZEN
- INVALID(frozen)_KEY – Request ASC/ASCQ 24h/05h for SECURITY WORKING KEY FROZEN

The third exception is an error that is otherwise indistinguishable from an invalid field value:

- NONCE_NOT_UNIQUE – Request ASC/ASCQ 24h/06h for NONCE NOT UNIQUE

In addition, an additional sense code not proposed above will be added because the potentially frequent occurrence of the condition makes the time require to parse the sense key specific data a burden on system performance:

– Request ASC/ASCQ 24h/07h NONCE TIMESTAMP OUT OF RANGE

11) Section 3 (Level 1 – Integrity of Capabilities) was incorporated incorrectly in OSD r07a as noted in comment IBM 7). This will be corrected.
    a) It will be noted that a zero in the Request Nonce CDB field specifies that the CDB has been constructed according to Level 1 rules.
    b) A detailed definition of the security token (aka ChannelId, see comment IBM 12) will be added to the subclause describing Level 1 as follows:
        - Make the security token a concatenation of the initiator identifier (see SAM-3) and the target identifier (see SAM-3), in that exact order;
        - Require that, if the SCSI transport protocol is capable of changing the initiator and target identifiers in an I_T Nexus without generating an I_T Nexus Loss event notification (see SAM-3), the security token shall be recomputed for every command sent;
        - Explain that if the initiator and target are in separate SCSI Domains (connected by a bridging device, presumably) the security token built by the initiator will be different from the security token built by the target, resulting in comparison failures of the integrity check values; and
        - Note that the problems caused by multiple SCSI Domains can be solved by using a different security level (e.g. security level 2).
    c) No changes other than those noted in other comments will be made.
12) Section 4 (Per Request Nonces for Level 2 and Level 3) will be used to update 4.6.5.4 (Security nonce format) as follows
    a) The first two bullets in the Section 4 hanging paragraph will be incorporated,
    b) Section 4.1 (Background) will not be incorporated directly because any requirements it may state should be covered as part of the incorporation of Section 4.7 (Additional Attributes on Partition Object),
    c) Section 4.2 (Requirements) will not be incorporated because its contents are not appropriate for inclusion in a T10 standard,
    d) The nonce format in 4.6.5.4 (Security nonce format) will be replaced with the format described in 4.3 (Structure of the Per Command Nonce),
    e) Section 4.4 (Use of Nonce for Anti Replay) will be combined with the new attributes defined in Security Nonce page (see below) to place requirements on device server validation of nonces.
    Notes: A) returning the current time as specified in a couple of error cases will be accomplished by

requiring the use of the descriptor sense data format and placement of the time in bytes 4-9 of the Command-specific information sense data descriptor.
B) the wording for the "big hammer" handling of requests too far in the future will be greatly modified in order to make it conform to T10 standardeze,

f) Section 4.5 (Host Protocol) will not be incorporated because it places requirements on application clients and the nature of those requirements are an obvious derivation of the duplicate nonce rejection requirements placed on the device server,

g) Section 4.6 (Use of Time) will not be incorporated because it contains statements that are in direct conflict with statements in Section 4.3 (Structure of the Per Command Nonce),

h) Section 4.7 (Additional Attributes on Partition Object) will be incorporated as a new Security Nonce attributes page. As an added aid to application clients, the new attributes page will include a copy of the clock attribute in the Root Information page. In this way, application clients can update all the information they need for nonce handling from one attributes page, also

i) Text in Section 4 not already mentioned above that is not appropriate to a T10 standard will be reworded or not included.

13) Section 5 (Level 2 – Integrity of Arguments) was incorporated incorrectly in OSD r07a as noted in comment IBM 14) and comment Editor 6). This will be corrected. No changes other than those noted in other comments will be made.

14) Section 6 (Level 3 – Integrity of Arguments and Data) is incorporated in OSD r07a and no memorable incorporation errors were reported other than those for Levels 1 and 2.

a) However, it has since come to light that the locations of the Data-In and Data-Out integrity check values (in the attributes and CDB, respectively) presents implementations problems. The following changes will be made:

- The DATA-OUT DIGITAL SIGNATURE field (aka integrity check value) will be removed from the CDB.
- The Data-In digital signature attribute will be removed from the Current Command attributes page.
- New buffer segments will be added to the end of the Data-In and Data-Out buffers (see 4.4). The new segments will be exactly the size of an integrity check value and will contain the DATA-IN INTEGRITY CHECK VALUE field and DATA-OUT INTEGRITY CHECK VALUE field, respectively. See also comment Editor 11).

b) No changes other than those noted in other comments will be made.

15) Information regarding the attacks, threats, and risks associated with each security level in Object Store Security Document will be incorporated to resolve comment EMC 4).

16) Although the Object Store Security Document fails to specifically define a Security attributes page for user objects, section 2.4 appears to require one. So, a Security attributes page will be added for user objects as described in comment Panasas 9).

17) Section 8.8 (Parameterizing Cryptographic Primitives) lacks sufficient details regarding how an "ordered and numbered list of primitives" are to be represented as attributes. Therefore, incorporation of this concept will be deferred to OSD-2.

18) Table 4 (OSD Specific Model Objects) will be updated to include both Credential and Capability references.

Document 03-279r1 has been uploaded to further clarify the incorporation of 03-279r0. The two documents are identical except that 03-279r1 contains PDF markups that will guide the incorporation of 03-279r0.

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 10 (References) will not been incorporated because they contain explanatory material that is not appropriate for inclusion in a T10 standard.
2) The Section 2 (OSD Grouping) hanging paragraph will not be incorporated except that it notes the need for a global search and replace of 'group' to 'partition'.
3) Section 2.1 (Partitions) will contribute a glossary entry and the definition of REMOVE OBJECT PARTITION behavior when user objects are still in the partition. Otherwise, it will not be incorporated.
4) The Section 2.2 (Collections) hanging paragraph, Section 2.2.1 (Representation), and 2.2.2 (Operations) will be used to:
   a) Add a new a,b,c list entry in 4.6.2.1 (Stored data object types),
   b) Update 4.6.2.5 (User objects),
   c) Create 4.6.2.6 (Collections) and update table 4 (OSD Specific Model Objects),
   d) Update tables 23 (Permissions bit mask format) & 24 (Capabilities Permission Bits),
   e) Add three new commands in clause 6 (Commands for OSD devices),
   f) Update table 55 (OSD Attribute Pages), and
   g) Create 7.1.2.15 (User Object Collections attributes page).

   See also the Section 4.1 (Attribute Pages) discussion below for related changes.
5) Section 3 (Attributes Terminology) will not be incorporated because the columns labeled 'May be set' and 'OSD Provided' in every attributes definition table already define the concepts. It is further noted, that the concepts in section 3 are not used anywhere else in the OSD Grouping and Attributes document.
6) The Section 4 (Attributes Overview) hanging paragraphs will not be incorporated because they are already covered in OSD r07a.
7) In Section 4.1 (Attribute Pages):
   a) Table 1 (Object Attribute Page Numbers) will be used to update table 8 (Object Attribute Page Numbers) and the P+, R+ text following table 8,
   b) A new C+ number will be added for collections attributes,
   c) Table 2 (Attribute Page Number Allocations) will be used to update table 9 (Object attribute page number sets),
   d) Collections Directory/Information/Timestamps attributes pages will be added with their default contents cloned from the equivalent User Object xxx attribute pages, then the used capacity attribute will be removed from the Information attributes page clone. A Resources attributes page will not be created because there are no attributes in the User Object Resources attributes page that make sense for a Collection,
   e) Otherwise, the text will not be incorporated because it is already covered in OSD r07a.

8) Section 4.2 [Attribute Type (How are Attributes Defined?)] will not be incorporated directly, but it will result in the following changes:

   a) In table 9 (Object attribute page number sets), the table footnotes will be removed and the last column labeled 'Assignment' will be replaced with two columns labeled 'Page Number Access Allowed' and 'Attribute List Access Allowed'. The row contents in both columns will be 'Yes' with two exceptions. In the 'Dynamically defined by applications' row, the 'Page Number Access Allowed' column will contain 'No'. In the 'Vendor specific' row both columns will contain 'n/a',

   b) The following paragraph will be added at the end of 7.1.1:

      A get attributes request for an attribute or attribute page having no previously established value shall not be considered an error. When an attribute value that has not been previously established is requested, a list entry format value (see 7.1.3.5) having zero in the attribute length field shall be returned. When an attribute page that has no established definition is requested, a null attributes page (see 7.1.2.xx) shall be returned.

   c) A new subclause 7.1.2.xx (Null Attributes Page) will be added as the last subclause in 7.1.2 to define an attributes page format containing only the page number field and the page length field with the page length field set to zero,

   d) Additional changes regarding this issue are described in the response to comment Panasas 13).

9) Based on section 5.2 (Operations to be Kept in the Spec with Modifications) all 7.1.3 discussion of accessing attributes belonging to other objects will be removed (see comment Seagate 39),

10) Section 5.3 (Combining Data and Attribute Operations) will be used to construct a new subclause 4.6.3.2 (Function ordering for commands that get and/or set attributes). The normative part of the new subclause will be the following list:

    The processing of commands other than REMOVE, REMOVE OBJECT PARTITION, and REMOVE OBJECT COLLECTION that include getting or setting attributes shall be performed in the following order:
    1) Processing those command functions not related to attributes;
    2) Process any set attributes operations specified in the CDB;
    3) Process any set attributes operations resulting from the processing of the command (e.g., changes due to a WRITE command); and
    4) Process any get attributes operations specified in the CDB.

    The processing of REMOVE, REMOVE OBJECT PARTITION, and REMOVE OBJECT COLLECTION commands that include getting or setting attributes shall be performed in the following order:
    1) Process any set attributes operations specified in the CDB;
    2) Process any get attributes operations specified in the CDB; and
    3) Processing those command functions not related to attributes.

11) 5.1.2.3 [Get attributes parameters (pages only)], 5.1.2.4 [Get attributes parameters (page and list)], 5.1.2.12 [Set attributes parameters (values only)], and 5.1.2.13 [Set attributes parameters (value and list)] will be updated to reflect the last paragraph of section 5.4 (CDB Parameters for Attributes). The remainder of section 5.4 will not be incorporated. **Note:** these changes affect the length of fields in the common CDB format.

12) Section 6 (Partitions, Collections, and Inheritance) will not be incorporated directly for the following reasons:

    a) The issues raised by the first paragraph are resolved by the response to comment Seagate 1), and

    b) No details are provided for resolving the issues raised by the second paragraph.

13) Section 7 (Object Type Attribute) will be incorporated as follows:
    a) The proposed new attribute will not be added because user objects and collections do not share a single Information attributes page in which to add it. The User Object Information attributes page (number 1h) is not the same as the Collections Information attributes page (number 6000 0001h), however
    b) 6.12 (LIST command) will be updated to restrict the object identifiers returned to user objects (not collections).
14) Section 8 (Additional Changes to Definitions of OSD Attribute Pages and Attribute Parameters) will be incorporated as follows:
    a) The first bullet matches comment Seagate 28) and will be incorporated as written,
    b) The second bullet will be incorporated as described in the response to comment Panasas 12), and
    c) The third (last) bullet will be incorporated as written.
15) Section 9 (Ideas for Next Version of the Standard: Temporary Objects, Garbage Collection, Expiration Date, Secure Erase, etc.) will not be incorporated because it discusses only concepts that have been deferred to OSD-2.

Document 03-280r3 has been uploaded to further clarify the incorporation of 03-280r1. The two documents are identical except that 03-280r3 contains PDF markups that will guide the incorporation of 03-280r1.

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.

Editor 8) Rename 4.2 to 'The OSD object abstraction' (Accepted, Editorial) [11]
page 12, 4.2, subclause title

As evidenced by comment IBM 2), the current 4.2 title 'Overall OSD Architecture' is leading to confusion wherein it is believed that the subclause constitutes part of the OSD Model. This is not correct and to avoid continuing that confusing the subclause title will be changed to 'The OSD object abstraction'.

Editor 9) Remove all discussion of OSD Policies (Accepted, Substantive) [12]
Global

Since none of the comments received (including all of the TWG documents) provides text defining OSD policies, all discussion of OSD policies will be removed. Documents going to T10 Letter Ballot cannot contain TBD material.

Editor 10) Do "Dirty Word" search (Accepted, Editorial) [13]
Global

Review for "dirty words" (e.g., can, will, must), improper use of 'such as', 'for example', and other T10 style guidelines as described in 01-313r2. Also 'when' should be 'if' in most cases, a T10 concern that postdates the style guide.

Editor 11) Add Data-In and Data-Out buffer offset information to the CDB (Accepted, Substantive) [14]
Multiple clauses

With the addition of yet another segment to both the Data-In and Data-Out buffers (see comment OSD TWG 2), the placement of bytes in these ever more complex structures can no longer be driven solely by the sizes of the fields in each segment. For each buffer segment not based on the zero offset in the buffer, a 32-bit offset value will be added to the CDB. Specifically, the following offsets will be added:

a) Data-In Get Attributes offset
b) Data-In Integrity Check Value offset
c) Data-Out Get Attributes List offset
d) Data-Out Set Attributes offset
e) Data-Out Integrity Check Value offset

Because the alignment of buffer segments by initiators will fall on memory management page boundaries, an exponent and mantissa format will serve to communicate the offset values most efficiently as follows:

a) Exponent (4 bits) - $2^{(Exponent+8)}$ (computed exponent range 256 to 8,388,608)
b) Mantissa (28 bits)

Thus, byte_offset = Mantissa * ($2^{(Exponent+8)}$)

For Exponent=0, the byte_offset range is 0 to 549,755,813,632 (7F FFFF FF00h).

Editor 12) Update Normative References (Accepted, Editorial) [15]
Clause 2

The normative references are out of date. SAM-2 is shown as in development. SPC-2 has no approval date. SAM-3 and SPC-3 are not even mentioned.

Editor 13) Expunge all Editors Notes (Accepted, Substantive) [16]
Global

After all other changes have been made, any remaining Editors Notes will be removed and any substantive or non-substantive changes described by those notes will be incorporated.

3. EMC Corp.

David Black from EMC Corp. submitted the following comments.

EMC 1) Encrypted connection required for Credential transmittal (No Action) [17]
page 23, 4.6.5.1

> 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.

Editor's Note: The requirement you are looking for appears in 4.3, in the second paragraph under figure 3. In OSD r07a, the requirement reads as follows:

> The communications mechanism used by the Security Manager shall be secure from all attacks and shall encrypt the data it transfers for data privacy.

Comment IBM 4) notes that this wording requires encrypted communications with both initiators and OSD devices whereas only the communications with initiators are required to be encrypted. See comment IBM 4) for the revised wording for the cited sentence that will appear in r08.

EMC 2) Add cross reference to Credential Format (Accepted, Editorial) [18]
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) [19]
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 (Accepted, Editorial) [20]
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.

Editor's Note: The information in the Object Store Security Document (see comment OSD TWG 2) pertaining to attacks/threats/risks will be incorporated in the security model subclauses. Section 4.4 of the Object Store Security Document appears to cover level 3 and replay attacks.

EMC 5) Extensible hash and encryption functions required (Accepted, Substantive) [21]
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.

Editor's Note: The MAC function capability field described in section 2.4 of the Object Store Security Document (see comment OSD TWG 2) appears to address this issue.

EMC 6) Nonce requirements inadequate (Accepted, Substantive) [22]
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 resolved 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 (Accepted, Editorial) [23]
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

Editor's Note: The following security terms will be added to the glossary:

- application client nonce
- capability
- capability nonce
- capability integrity check value
- credential
- integrity check value
- nonce
- secret key
- security manager
- security token

IBM 2) Should 'Overall Architecture' mention security? (Rejected) [24]
pages 12-13, 4.2

Should this section mention security?

Reason for Rejection: The purpose of subclause 4.2 is to compare and contrast the traditional storage access model and the OSD model. Both models provide for security and the differences in how security is handled are not substantially different from the differences in which addressing is handled.

Adding a discussion of security to 4.2 would obfuscate the main topic that needs to be addressed.

Comment Editor 8) changes the title of 4.2 in an attempt to reduce the confusion that caused this comment.

IBM 3) Security Manager should not be optional (Rejected) [25]
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 (Accepted, Substantive) [26]
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.

Editor's Note: Change from:

> The communications mechanism used by the Security Manager shall be secure from all attacks and shall encrypt the data it transfers for data privacy. The Security Manager may reside in the OSD storage devices or

in initiators, but the security requirements on the communications mechanism are not altered in such configurations.

to:

When sending credentials to an application client, ~~The communications mechanism used by~~ the Security Manager shall use a communications mechanism that is ~~be~~ secure from all attacks and shall encrypt the data it transfers for data privacy. The Security Manager may reside in the OSD storage devices, ~~or~~ in initiators, or as a separate entity, but the security requirements on the communications mechanism shall not change based on the location of the Security Manager ~~are not altered in such configurations~~.

IBM 5) 'command key' s/b 'capability key' (Accepted, Editorial) [27]
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.

Editor's Note: Actually, the change to 'capability key' needs to be implemented throughout the working draft.

IBM 6) Security Level 1 definition incorrect (Accepted, Editorial) [28]
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)

Editor's Note: Change from:

The application client requests credentials and command keys from the security manager for OSD operations it needs to preform and sends those credentials to the OSD as part of commands that are digitally signed with the command key.

to:

The application client requests credentials and command keys from the security manager for OSD operations it needs to preform and sends those credentials to the OSD as part of commands that ~~are~~ may be digitally signed with the command key.

IBM 7) Recombine security Levels 1 and 2 as per the SNIA OSD TWG description (Accepted, Substantive) [29]
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) [30]
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) [31]
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) [32]
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) [33]
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) [34]
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) [35]
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) [36]
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) [37]
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) [38]
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) [39]
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) [40]
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) [41]
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) [42]
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) [43]
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) [44]
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) [45]
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 (Accepted, Substantive) [46]
page 36, 5.1.2.1

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

Editor's Note: This comment will be resolved as described in the response to comment Panasas 21).

IBM 25) Request more Additional Sense Code assignments (Rejected) [47]
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) [48]
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) [49]
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) [50]
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) [51]
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) [52]
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) [53]
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) [54]
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 (Deferred to OSD-2) [55]

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).

Editor's Note: As described in section 9 of the OSD Attributes Draft (see comment OSD TWG 3), attribute templates have been deferred to OSD-2.

The fate of the CREATE ATTRIBUTES PAGE command will be resolved as described in the response to comment Seagate 19).

Panasas 2) Attribute and Data Ordering (Accepted, Substantive) [56]
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

Editor's Note: The ordering of segments within the Data-In and Data-Out buffers will be coordinated with the order that OSD Grouping and Attributes document (see comment OSD TWG 3) specifies for the processing of data and attributes. This will facilitate implementation of OSD on SCSI transport protocols that do not support modify data pointers.

Panasas 3) Atomicity of Set Attributes writes (Accepted, Substantive) [57]
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.

Editor's Note: This comment will be resolved as described in section 5.3 of the OSD Attributes Draft. See comment OSD TWG 3) for additional discussion of the resolution.

Panasas 4) Root & Partition Attribute Directory Contents (Deferred to OSD-2) [58]
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.

Editor's Note: Although the issue raised by the comment is very real, there is no consensus in favor of making the proposed change within the SNIA OSD TWG.

Panasas 5) Size of Object Attribute Name Space (Accepted, Substantive) [59]
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.

Editor's Note: This comment will be resolved as described in section 4.1 of the OSD Attributes Draft. It appears that some, if not all, the requested changes are present there. See comment OSD TWG 3) for additional discussion of the resolution.

Panasas 6) Large Capability and Where To Store It (Deferred to OSD-2) [60]
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 (Accepted, Substantive) [61]
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.

Editor's Note: In general, this comment will be resolved as described in the response to comment Seagate 39). The last paragraph of the comment provides interesting guidance of resolving the general issue. However, it appears that the cited table should be removed completely in preference for the table in the preceding subclause.

Panasas 8) Get and Set Attribute Parameters (Accepted, Substantive) [62]
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 (Accepted, Substantive) [63]
New subclause

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.

Editor's Note: The Object Store Security Document (see comment OSD TWG 2) fails to specifically define a Security attributes page for user objects but section 2.4 appears to require one. So, acceptance of this comment will be treated as an addendum to the Object Store Security Document.

Panasas 10) CREATE CDB missing reserved bytes (Accepted, Editorial) [64]
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.

Editor's Note: The reserved bytes for the LENGTH and STARTING BYTE ADDRESS fields are not shown in the CREATE command CDB. This is the cause of the 28 byte REQUESTED USER_OBJECT_ID field. The reserved bytes will be restored.

The get/set attributes issues described in the second paragraph will be resolved as described in comment OSD TWG 3).

Panasas 11) Root object "Used Capacity" attribute (Accepted, Substantive) [65]
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.

Editor's Note: This comment will be resolved as described in section 8 of the OSD Attributes Draft. See comment OSD TWG 3) for additional discussion of the resolution.

Panasas 12) Root object "OSD Security Level" attribute (Accepted, Substantive) [66]
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: The following changes will be made:

a) The Root Information attribute called "OSD security level" will be renamed to "Minimum security level", moved to a newly created Security attributes page, and will be specified to affect accesses to the Root object;
b) A new Root Security attribute called "Partition minimum security level" will be defined and described as providing the default minimum security level for newly created partitions;
c) A new Group (aka Partition) Security attribute called "Minimum security level" will be defined and described as affecting accesses to the partition and user objects in the partition; and
d) A new Root Security attribute called "Maximum supported security level" will be defined and described as indicating the highest security level supported by the OSD.

Panasas 13) Group and User-object Information Attributes Page (Accepted, Editorial) [67]
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.

Editor's Note: Using the Root Resources attributes page as an example, table 62 defines the format of attribute data if the entire page is requested without recourse to the FFFF FFFFh attribute number. The first part of the statement says that no such facility is provided for the Root, Group (aka Partition), and User Information attributes pages. The decision not to provide a page format is based on the presence of variable length attributes in the page.

Because no page format is defined, it is illegal to reference these pages without including an attribute number value. The second part of the statement attempts to covey the necessary requirement.

However, the second part of the cited text fails to clearly identify the page-only reference case. Furthermore, changes resulting from comment OSD TWG 3) provide that a get attributes should always succeed, regardless of page format definition. Therefore, for all pages that do not have a defined page format, the error definition text will be revised as follows:

No page format is specified for … If a CDB ~~get or~~ set attributes field specifies the page number of the … attributes page and no attribute number, 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.

Panasas 14) Command with GetAttr and/or SetAttr and execution ordering (Accepted, Substantive) [68]
new text

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.

Editor's Note: This comment will be resolved as described in section 5.3 of the OSD Attributes Draft. See comment OSD TWG 3) for additional discussion of the resolution.

Panasas 15) Constraints on number of objects (Accepted, Substantive) [69]
various pages and clauses
see also: comment Seagate 33)

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.

Editor's Note: The number of objects in a partition issue will be resolved as described in the ObS Identifying Objects document. See comment OSD TWG 1) for additional discussion of the resolution.

The following changes will be made to address the other issues raised by this comment:

a) The attribute number for the "user objects per group" Root Resources attributes page will be changed to reserved; and
b) The "object count" attribute size in the Group (aka Partition) Resources attributes page will be changed from 4 to 8 bytes (this change also proposed in comment Seagate 33).

Panasas 16) Last-Access time (Deferred to OSD-2) [70]
New subclause in CDB definition

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).

Editor's Note: This comment will be resolved as described in the response to comment Panasas 23).

Panasas 17) Remove "Remaining Capacity" attribute(s) (Accepted, Substantive) [71]
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.

Editor's Note: The capacity quota and used capacity attributes are already defined in OSD r07a. So the only change needed is removing the remaining capacity attribute.

Panasas 18) Terminology (Accepted, Editorial) [72]
pages 5-8, 3.1 & 3.2

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?

Editor's Note: This comment will be resolved as described in the response to comment IBM 1).

The entire list of Capability component fields will not be included in the glossary because that is inappropriate for a T10 standard. The term MAC will be used as little as possible and not appear in the glossary because from a T10 perspective it is confusing jargon.

Panasas 19) Credential Format (Accepted, Substantive) [73]
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 (Rejected) [74]
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.

Reason for Rejection: The capability fields defined in section 2.4 of the Object Store Security Document (see comment OSD TWG 2) will be incorporated. Fields in OSD r07a that are not in section 2.4 of the Object Store Security Document will be removed.

Of the fields listed in this comment, Boot epoch and Minimum Security do not appear in section 2.4 of the Object Store Security Document and thus will not be incorporated. The Hash Algorithm Used field defined by section 2.4 of the Object Store Security Document is 4 bits not 16. The Audit Field is included in the Nonce, not vice versa. The Credential Create Time field is discussed in comment Panasas 24).

Panasas 21) Accessing attributes in other objects deferred; add more Permissions Bits (Accepted, Substantive) [75]
Global, page 37, 5.1.2.1.1, table 23
see also: comment IBM 24)

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.

Editor's Note: Getting and setting attributes in an object other than the object directly addressed in the CDB has been deferred to OSD-2.

So that there are enough permissions bits for all future uses (including those suggested by this comment) the size of the PERMISSIONS BIT MASK field in the Capability will be increased by 2 to 4 bytes.

Panasas 22) Key Version field too large (Accepted, Substantive) [76]
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.

Editor's Note: Section 2.4 of the Object Store Security Document (see comment OSD TWG 2) sets the Key Version field size at 4 bits.

Panasas 23) Timestamp bypass (Deferred to OSD-2) [77]
New subclause in CDB definition
see also: comment Panasas 16)

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 (Accepted, Substantive) [78]
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.

Editor's Note: Section 2.4 of the Object Store Security Document (see comment OSD TWG 2) includes a Creation Time field in the capability that is clearly defined as the creation time of the object being accessed. It appears that the editor was confused when incorporating the previous version of the Object Store Security Document.

Panasas 25) Credential Nonce (Rejected) [79]
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.

Reason for Rejection: Section 2.4 of the Object Store Security Document (see comment OSD TWG 2) includes the audit field in the credential nonce instead of vice versa.

6. Seagate Technology

Sami Iren from Seagate Technology submitted the following comments.

Seagate 1) Attribute inheritance wording is unclear (Accepted, Editorial) [80]
Page 17, 4.6.2.1, list entries b) & c)

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.

Editor’s Note: The cited sentences will be changed to:

> b) … The default ~~attributes~~ values for ~~a group~~ some partition object attributes are copied (i.e., inherited) from ~~the~~ specified attributes in the root object. …
>
> c) … Default ~~attributes~~ values for ~~a~~ some user object attributes are copied (i.e., inherited) from ~~the~~ specified attributes of the group object in which it is listed. …

Seagate 2) Last attribute number is FFFF FFFEh (Accepted, Editorial) [81]
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 (Rejected) [82]
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)

Reason for Rejection: The latest revision of the OSD Attributes Draft (see comment OSD TWG 3) does not describe the removal of attribute directories. In fact, the third bullet in Section 8 (Additional Changes to Definitions of OSD Attribute Pages and Attribute Parameters) specifically mentions renaming the attribute directory pages.

Seagate 4) Defer sessions to OSD-2 (Accepted, Substantive) [83]
Global

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?

Editor’s Note: Sessions have been deferred to OSD-2. All discussion of sessions currently in the working draft will be removed.

Seagate 5) 'a' s/b 'an' (Accepted, Editorial) [84]
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 (Accepted, Editorial) [85]
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.

Editor’s Note: In figure 4, 'OSD Storage Device' will be changed to 'OSD Device Server'.

In the first paragraph after the a,b,c list after figure 4, the second sentence will be changed from:

> An application client that has just the credential (e.g., obtained by monitoring service delivery subsystem interactions) but …

to

> An application client that has just the ~~credential~~ capability (e.g., obtained by monitoring ~~service delivery subsystem interactions~~ CDBs sent to the OSD) but …

Seagate 7) Remove 'that' & insert 'is' (Accepted, Editorial) [86]
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) [87]
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) [88]
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 (Accepted, Editorial) [89]
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.

It will be changed to read:

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

Seagate 11) Insert cross reference to table 15 (Accepted, Editorial) [90]
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) [91]
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) [92]
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) OSD example needed (Accepted, Editorial) [93]
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. However owing to the newness of OSD, an informative annex containing this information may be acceptable to T10. Therefore the following changes will be made:

a) The current content of 4.7.1 and 4.7.4 will be moved to a new informative annex (Annex B);
b) Subclause 4.7.2 will be removed as per comment Seagate 13) and subclause 4.7.3 will be removed as per the editor’s note in OSD r07a;
c) The issues with the current content will be resolved by the editor to the best of the editor’s ability; and
d) Any additions to the new annex received before 1 September will be added to the new annex provided they received information is consistent with the format found in 4.7.4

Seagate 15) Linked Command Support (No Action) [94]
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) [95]
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) [96]
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) [97]
page 53, 6.5, table 44

The last row should read 235, not 145.

Seagate 19) Remove the CREATE ATTRIBUTES PAGE command (Accepted, Substantive) [98]
pages 55-56, 6.6
see also: comment Panasas 1)

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

Editor's Note: As described in section 9 of the OSD Attributes Draft (see comment OSD TWG 3), attribute templates and dynamic creation attribute pages containing references to other attributes have been deferred to OSD-2.

CREATE ATTRIBUTES PAGE and REMOVE ATTRIBUTES PAGE will be removed from the working draft and no new commands will be added.

Seagate 20) Define FLUSH PARTITION & FLUSH OSD (Deferred to OSD-2) [99]
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) Defer IMPORT USER OBJECT command to OSD-2 (Accepted, Substantive) [100]
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. Therefore the group has advised the editor via email that the definition of an IMPORT USER OBJECT command must be deferred to OSD-2.

Seagate 22) Increase Index field size in LIST CDB (Accepted, Substantive) [101]
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 (Deferred to OSD-2) [102]
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

Editor's Note: As described in section 9 of the OSD Attributes Draft (see comment OSD TWG 3), attribute templates have been deferred to OSD-2. There are 8 reserved bytes in the current OPEN CDB format that may be used for session template id in OSD-2.

Seagate 24) Reduce WRITE byte count to 32 bits (Rejected) [103]
page 81, 6.19, table 54

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

Reason for Rejection: The CDB bytes are shared with the CAPACITY QUOTA field in the CREATE OBJECT CDB and the FORMATTED CAPACITY field in the FORMAT OSD CDB, so a reduction in this field size will not produce a reduction in the overall CDB size.

Another matter to consider is the chance that limiting data transfers to 4 GB is not sufficiently foresighted for OSD.

The total size of a single data transfer operation should not bother OSD device servers since all SCSI transport protocols allowed to targets to segment transfer operations as they see fit. On the initiator side, the transfer size will be self limiting because all bytes to be transferred must be locked in physical memory for HBA access.

As a last resort, OSD device servers that wish to limit the number of bytes send in a single WRITE may reject the command with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID FIELD IN CDB. As noted in the response to comment IBM 11), the sense key specific data may be set to identify the field cause the OSD device server heartburn.

Seagate 25) Increase Object Logical Length to 96 bits (No Action) [104]
pages 93-95, 7.1.2.11
see also: comment Seagate 35)

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

Editor's Note: Comment withdrawn by author.

Seagate 26) Limit access available to list format attributes (Accepted, Substantive) [105]
Page 83, 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.

Seagate 27) Remove directory pages and definitions (Rejected) [106]
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.

Reason for Rejection: The latest revision of the OSD Attributes Draft (see comment OSD TWG 3) does not describe the removal of attribute directories. In fact, the third bullet in Section 8 (Additional Changes to Definitions of OSD Attribute Pages and Attribute Parameters) specifically mentions renaming the attribute directory pages.

Seagate 28) Root 'Used Capacity' attribute should count all user objects (Accepted, Substantive) [107]
page 87, 7.1.2.6, 2nd p from bottom of pg
see also: comment OSD TWG 3)

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) [108]
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) [109]
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) [110]
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) What attribute reports the space actually used? (Rejected) [111]
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.

Reason for Rejection: As noted in comment Seagate 35), the Object Logical Length attribute provides this information as part of the User Object Resources attributes page. The confusion comes from the Object Logical Length attribute being in a different attributes page than the Used Capacity attribute. A resolution for this confusion is proposed in comment Seagate 35).

Seagate 33) Increase Group Count size from 4 to 8 bytes (Accepted, Substantive) [112]
pages 90-91, 7.1.2.9

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

Editor's Note: This comment will be resolved as described in the response to comment Panasas 15).

Seagate 34) 'total capacity' s/b 'remaining capacity' (Accepted, Substantive) [113]
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 (Accepted, Substantive) [114]
pages 89-90, 7.1.2.8 & pages 93-95, 7.1.2.11
see also: comment Seagate 32)

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) [115]
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) [116]

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) [117]

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) [118]

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.

The last paragraph of comment Panasas 7) provides additional guidance regarding changes in 7.1.3. However, it appears that the cited table should be removed completely in preference for the table in the preceding subclause.