Document Identifier: DSP0286

Date: 2024-08-13

Version: 1.0.0WIP90

SPDM to Storage Binding Specification

Information for Work-in-Progress version:

IMPORTANT: This document is not a standard. It does not necessarily reflect the views of DMTF or its members. Because this document is a Work in Progress, this document may still change, perhaps profoundly and without notice. This document is available for public review and comment until superseded.

Provide any comments through the DMTF Feedback Portal: https://www.dmtf.org/standards/feedback

Supersedes: None

Document Class: Normative

Document Status: Work in Progress

Document Language: en-US

11 DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems management and interoperability. Members and non-members may reproduce DMTF specifications and documents, provided that correct attribution is given. As DMTF specifications may be revised from time to time, the particular version and release date should always be noted.

12 Implementation of certain elements of this standard or proposed standard may be subject to third-party patent rights, including provisional patent rights (herein "patent rights"). DMTF makes no representations to users of the standard as to the existence of such rights, and is not responsible to recognize, disclose, or identify any or all such third-party patent right owners or claimants, nor for any incomplete or inaccurate identification or disclosure of such rights, owners, or claimants. DMTF shall have no liability to any party, in any manner or circumstance, under any legal theory whatsoever, for failure to recognize, disclose, or identify any such third-party patent rights, or for such party's reliance on the standard or incorporation thereof in its product, protocols, or testing procedures. DMTF shall have no liability to any party implementing such standard, whether such implementation is foreseeable or not, nor to any patent owner or claimant, and shall have no liability or responsibility for costs or losses incurred if a standard is withdrawn or modified after publication, and shall be indemnified and held harmless by any party implementing the standard from any and all claims of infringement by a patent owner for such implementations.

13 For information about patents held by third parties which have notified DMTF that, in their opinion, such patents may relate to or impact implementations of DMTF standards, visit https://www.dmtf.org/about/policies/disclosures.

14 This document's normative language is English. Translation into other languages is permitted.

15 CONTENTS

1 Foreword . . . 5
1.1 Acknowledgments . . . 5
2 Introduction . . . 6
2.1 Document conventions . . . 6
2.2 Other conventions . . . 6
3 Scope . . . 7
3.1 Out of scope . . . 7
3.2 Normative references . . . 7
3.3 Terms and definitions . . . 7
3.3.1 Equivalent terms . . . 8
3.4 Symbols and abbreviated terms . . . 9
3.5 Binding Information . . . 9
3.6 Annotation of differences between storage protocols . . . 9
3.7 Bit and byte ordering . . . 9
4 Theory of operation . . . 10
5 Security protocol commands . . . 11
5.1 Protocol command format . . . 11
5.1.1 IF-SEND format . . . 11
5.1.2 IF-RECV format . . . 11
5.1.3 Namespace addressing . . . 12
5.2 Command management . . . 12
6 SPDM Storage commands . . . 14
6.1 SPDM Storage Discovery . . . 14
6.2 SPDM Storage Pending Info . . . 15
6.3 SPDM Storage Message . . . 17
6.3.1 SPDM Storage Message IF-SEND . . . 18
6.3.2 SPDM Storage Message IF-RECV . . . 18
6.3.3 SPDM Storage Message status . . . 18
6.3.4 Encapsulated request flow . . . 19
6.4 SPDM Storage Secured Message . . . 19
6.4.1 Use of descriptors . . . 22
7 Storage protocol specific behavior . . . 24
7.1 Transcript hash calculation . . . 24
8 Storage specific accommodations . . . 25
8.1 Device reset handling . . . 25
8.2 Status response hierarchy . . . 25
8.2.1 SPDM Storage protocol status . . . 26
8.2.1.1 SCSI protocol status . . . 26
8.2.1.2 ATA protocol status . . . 26
8.2.1.3 NVMe protocol status . . . 27
8.2.2 Secured message encapsulated status . . . 27

8.2.2.1 Vendor defined secured message encapsulated status 28
8.3 Padded transactions 30
8.4 Multi-path handling 30
9 ANNEX A (informative) change log 31
9.1 Version 1.0.0 (in progress) 31
10 Bibliography 32

16 # 1 Foreword

17 The Security Protocols and Data Models (SPDM) Working Group prepared the *SPDM to Storage Binding Specification* (DSP0286).

18 DMTF is a not-for-profit association of industry members that promotes enterprise and systems management and interoperability. For information about the DMTF, see https://www.dmtf.org.

19 ## 1.1 Acknowledgments

20 The DMTF acknowledges the following individuals for their contributions to this document:

2 Introduction

This specification binds SPDM messages (DSP0274) and SPDM Secured Messages (DSP0277) to storage protocols.

2.1 Document conventions

- Document titles appear in *italics*.
- The first occurrence of each important term appears in *italics* with a link to its definition.
- ABNF rules appear in a monospaced font.

2.2 Other conventions

Unless otherwise specified, all figures are informative.

26 # 3 Scope

27 This document defines the format of Security Protocol and Data Model (SPDM) messages over storage protocols.

28 ## 3.1 Out of scope

29 The following topics are out of scope for this specification:

- Asynchronous notification from the Responder to the Requester.
- Translation of commands between different storage protocols.

30 ## 3.2 Normative references

31 The following referenced documents are indispensable for the application of this specification. For dated or versioned references, only the edition cited (including any corrigenda or DMTF update versions) applies. For references without a date or version, the latest published edition of the referenced document (including any corrigenda or DMTF update versions) applies.

- ACS-4, ATA/ATAPI Command Set — 4, ISO/IEC 17760-104, https://www.iso.org/standard/83120.html
- DMTF DSP0274, *Security Protocol and Data Model (SPDM) Base Specification*, https://www.dmtf.org/dsp/DSP0274
- DMTF DSP0277, *Secured Messages using SPDM Specification*, https://www.dmtf.org/dsp/DSP0277
- *ISO/IEC Directives, Part 2, Principles and rules for the structure and drafting of ISO and IEC documents — 2021 (9th edition)*
- NVMe, NVM Express Base Specification, https://nvmexpress.org/wp-content/uploads/NVM-Express-Base-Specification-2.0c-2022.10.04-Ratified.pdf
- IETF RFC5234, *Augmented BNF for Syntax Specifications: ABNF*, January 2008, https://tools.ietf.org/html/rfc5234
- SAM-6, SCSI Architectural Model — 6, ANSI INCITS 546-2022, https://standards.incits.org/apps/group_public/project/details.php?project_id=1783
- SBC-4, SCSI Block Commands — 4, ANSI INCITS 506-2021, https://standards.incits.org/apps/group_public/project/details.php?project_id=1780
- SPC-6, SCSI Primary Commands — 6 rev 08, (INCITS number not assigned), http://www.t10.org/cgi-bin/ac.pl?t=f&f=spc6r08.pdf

32 ## 3.3 Terms and definitions

33 In this document, some terms have a specific meaning beyond the normal English meaning. This clause defines those terms.

34 The terms "shall" ("required"), "shall not," "should"("recommended"), "should not" ("not recommended"), "may," "need not" ("not required"), "can" and "cannot" in this document are to be interpreted as described in ISO/IEC Directives, Part 2, Clause 7. The terms in parentheses are alternatives for the preceding term, for use in exceptional cases when the preceding term cannot be used for linguistic reasons. Note that ISO/IEC Directives, Part 2, Clause 7 specifies additional alternatives. Occurrences of such additional alternatives shall be interpreted in their normal English meaning.

35 The terms "clause," "subclause," "paragraph," and "annex" in this document are to be interpreted as described in ISO/IEC Directives, Part 2, Clause 6.

36 The terms "normative" and "informative" in this document are to be interpreted as described in ISO/IEC Directives, Part 2, Clause 3. In this document, clauses, subclauses, or annexes labeled "(informative)" do not contain normative content. Notes and examples are always informative elements.

37 The terms that DSP0274 defines also apply to this document.

Term	Definition
IF-SEND	Generic term for a security related command that transfers data from the initiator to the target. For NVMe, this is Security Send. For SAS, this is SECURITY PROTOCOL OUT. For SATA, this is TRUSTED SEND and TRUSTED SEND DMA.
IF-RECV	Generic term for a security related command that transfers data from the target to the initiator. For NVMe, this is Security Receive. For SAS, this is SECURITY PROTOCOL IN. For SATA, this is TRUSTED RECEIVE and TRUSTED RECEIVE DMA.

38 ### 3.3.1 Equivalent terms

39 This binding specification primarily uses SPDM terminology. The following table explains how these terms align with terms from the underlying storage protocol specifications.

SPDM Binding Specification Term	SCSI Term	ACS Term	NVMe Term
IF-SEND	SECURITY PROTOCOL OUT	TRUSTED SEND or TRUSTED SEND DMA	Security Send
IF-RECV	SECURITY PROTOCOL IN	TRUSTED RECEIVE or TRUSTED RECEIVE DMA	Security Receive
Requester	SCSI host	Host	Host
Responder	SCSI target device	Device	Controller
Storage protocol	Service delivery subsystem	SATA protocol	NVMe protocol

3.4 Symbols and abbreviated terms

The abbreviations or notations defined in DSP0274 apply to this document.

3.5 Binding Information

This version of this specification binds to **all** published versions of the *Security Protocol and Data Model (SPDM) Specification* (DSP0274), though some functionality might not be available under all versions.

This version of this specification binds to these versions of the *Secured Messages using SPDM Specification* (DSP0277):

- Version 1.0.0 and all 1.0 errata versions
- Version 1.1.0 and all 1.1 errata versions
- Version 1.2.0 and all 1.2 errata versions

3.6 Annotation of differences between storage protocols

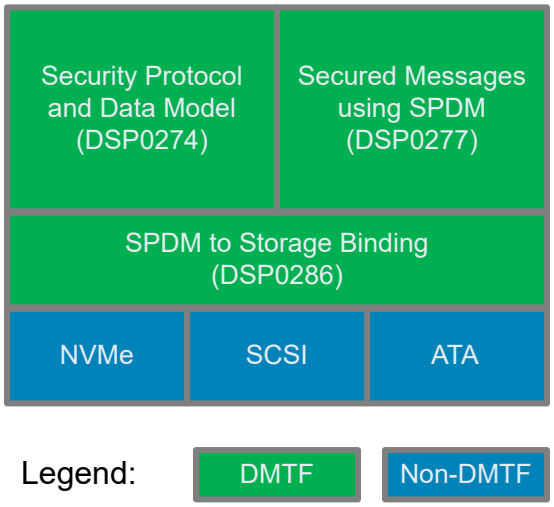
Where differences are not noted, this specification applies to all storage protocols that are in scope.

3.7 Bit and byte ordering

48 # 4 Theory of operation

49 **Figure 3422 — SPDM storage stack**

50



51 Figure 3422 — SPDM storage stack shows a high-level view of the SPDM to Storage Binding. The SPDM to Storage Binding allows endpoints to send messages defined by *Security Protocol and Data Model (SPDM) Specification* (DSP0274) between storage endpoints. Further, this specification allows endpoints to exchange messages using the *Secured Messages using SPDM Specification* (DSP0277).

52 # 5 Security protocol commands

53 SPC-6 specifies that the SECURITY PROTOCOL Code of `0xE8` shall be used for `IF-SEND` and `IF-RECV` commands for SPDM communication. Devices that use the SPDM SECURITY PROTOCOL Code shall conform to this specification.

54 ## 5.1 Protocol command format

55 ### 5.1.1 IF-SEND format

56 Table 1 — IF-SEND field definitions describes the use of fields in the protocol specific `IF-SEND` command. The fields listed map directly to the IF-SEND storage command, so the storage specification in use defines the field sizes and offsets. The use of fields not defined in the following table shall conform to the behavior defined in the protocol specification in use.

57 **Table 1 — IF-SEND field definitions**

Field	Usage
SECURITY PROTOCOL	This field shall be set to `0xE8` for SPDM commands.
SECURITY PROTOCOL SPECIFIC	This field shall be used for Command management.
INC_512	If allowed, devices shall support operation with either state of this field.
TRANSFER LENGTH	This field shall be the length of the command buffer to be transferred from the Requester to the Responder, inclusive of any required padding.

58 ### 5.1.2 IF-RECV format

59 Table 2 — IF-RECV field definitions describes the use of fields in the protocol specific `IF-RECV` command. The fields listed map directly to the IF-SEND storage command, so the storage specification in use defines the field sizes and offsets. The use of fields not defined in the following table shall conform to the behavior defined in the protocol specification in use.

60 **Table 2 — IF-RECV field definitions**

Field	Usage
SECURITY PROTOCOL	This field shall be set to `0xE8` for SPDM commands.
SECURITY PROTOCOL SPECIFIC	This field shall be used for Command management.

Field	Usage
INC_512	If allowed, devices shall support operation with either state of this field.
ALLOCATION LENGTH	This field shall be the length of the command buffer to receive data from the Responder to the Requester, inclusive of any required padding.

61 ### 5.1.3 Namespace addressing

62 When the `SECURITY PROTOCOL` field is set to `0xE8` for an NVMe device, the use of the Namespace Identifier (NSID) field shall be reserved. When the `SECURITY PROTOCOL` field is set to `0xE8` for a SCSI device, the use a LOGICAL UNIT NUMBER other than LUN 0 or the SECURITY PROTOCOL well-known logical unit shall be reserved. ATA does not support a field that is like Namespace Identifier or LOGICAL UNIT NUMBER.

63 ## 5.2 Command management

64 The `CommandManagement` structure shall be used in the SECURITY PROTOCOL SPECIFIC field in the `IF-SEND` and `IF-RECV` command descriptor blocks, as Table 3 — Command management fields.

65 **Table 3 — Command management fields**

Byte offset	Field	Size (bytes)	Description
0	Operation	1	Bit [1:0]. Shall contain the `ConnectionID` for the command. Bit [7:2]. Shall contain the `SPDMOperation` field, as Table 4 — SPDM operation codes defines.
1	Reserved	1	Reserved.

66 The `ConnectionID` field shall identify the SPDM connection for a command. All devices shall support `ConnectionID` 0, and shall support `ConnectionID`s from 0 to the value of the `MaxConnectionID` field in the SPDM Storage Discovery response.

67 The `SPDMOperation` field shall identify the SPDM Storage Operation Code for this command, as Table 4 — SPDM operation codes defines.

68 Some `SPDMOperation`s only support `IF-SEND` or `IF-RECV`, as Table 4 — SPDM operation codes defines.

69 **Table 4 — SPDM operation codes**

`SPDMOperation` Value	Command	Mandatory	`IF-SEND` Support	`IF-RECV` Support	Description
0x01	SPDM Storage Discovery	Mandatory	No	Yes	See SPDM Storage Discovery.

`SPDMOperation` Value	Command	Mandatory	`IF-SEND` Support	`IF-RECV` Support	Description
0x02	SPDM Storage Pending Info	Optional	No	Yes	See SPDM Storage Pending Info.
0x05	SPDM Storage Message	Mandatory	Yes	Yes	See SPDM Storage Message.
0x06	SPDM Storage Secured Message	Optional	Yes	Yes	See SPDM Storage Secured Message.
All other values	Reserved				Reserved.

70 # 6 SPDM Storage commands

71 The following commands are defined by this specification to manage the SPDM characteristics of the storage protocol.

72 ## 6.1 SPDM Storage Discovery

73 The SPDM Storage Discovery command reads the SPDM parameters from a device, and is read using `IF-RECV`. The SPDM Storage Discovery command shall be formatted as Table 5.1 — SPDM Storage Discovery command format defines.

74 **Table 5.1 — SPDM Storage Discovery command format**

IF-RECV Field	Value
`SPDMOperation`	0x1

75 **Table 5 — SPDM Storage Discovery response data**

Byte offset	Field	Size (bytes)	Description
0	DataLength	2	The `DataLength` field shall return the number of available bytes in the SPDM Storage Data, not including any Pad fields. This value might exceed the Allocation Length in the `IF-RECV` command.
2	StorageBindingVersion	2	The `StorageBindingVersion` for devices that implement this version of the SPDM to Storage Binding Specification shall be set to `0x1000`. Table 6 — Storage binding version format defines the format of this field.
4	Byte4	1	Bit [1:0]. MaxConnectionID. The `MaxConnectionID` field shall contain the maximum `ConnectionID` for the device. A value of 0 indicates that the device supports one connection. The `MaxConnectionID` and `ConnectionID` fields are specific to the device and storage protocol in use, and should not apply to other protocols. Bit [7:2]. Reserved.
5	Reserved	3	Reserved.

Byte offset	Field	Size (bytes)	Description
8	SupportedOperations	8	The `SupportedOperations` field shall return a bit mask of the `SPDMOperation`s that the Responder supports. The bit position corresponding to an `SPDMOperation` enumeration shall be set to `1` to indicate support for corresponding `SPDMOperation`, for both Mandatory and Optional `SPDMOperation`s. For instance, if a Responder supports `SPDM Storage Message`, it would set Bit[5] of `SupportedOperations` to `1`.
16	Reserved	16	Reserved.

76 **Table 6 — Storage binding version format**

Bit	Field	Value
[15:12]	MajorVersion	Shall be the major version of the storage protocol binding. See DSP0274 for description of major version.
[11:8]	MinorVersion	Shall be the minor version of the storage protocol binding. See DSP0274 for description of minor version.
[7:4]	UpdateVersionNumber	Shall be the update version of the storage protocol binding. See DSP0274 for description of Update version.
[3:0]	Alpha	Shall be the alpha version of the storage protocol binding. For released versions, this field shall be zero. See DSP0274 for description of alpha version.

77 ## 6.2 SPDM Storage Pending Info

78 The `SPDM Storage Pending Info` command returns information about pending response data being held by the

endpoint for the connection indicated in the requested `ConnectionID`, and is read using `IF-RECV`. The `SPDM Storage Pending Info` command shall only be used with `SPDM Storage Message` and `SPDM Storage Secured Message`. In storage protocols, the Requester must allocate a buffer to be used with the `IF-RECV` command, but does not know the size of the response data that the Responder has pending. The Requester can use one or more of the following approaches to manage the size of the `IF-RECV` buffer for response data. The format of the response data is fixed for a given version of the *SPDM to Storage Binding Specification*, so the Requester can reliably know the expected length of the response data.

- Use a lookup table or other similar mechanism to predict the size of the response data.
- If both endpoints support SPDM Large Messages (`CHUNK_CAP` = `1`), the Requester can allocate a receive buffer of at least `DataTransferSize`.
- If the Responder supports the `SPDM Storage Pending Info` message, the Requester can use the response data to ensure that it allocates a large enough buffer.

79 If a Responder uses the SPDM Large Message transfer mechanism to break a response into pieces, the `SPDM Storage Pending Info` response data does not update during the Large Message transfer. When the currently pending response has completed transmission, the Responder shall clear the `SPDM Storage Pending Info` response `ValidResponse` flag until a new command is received.

80 **Table 7 — SPDM Storage Pending Info command format**

IF-RECV Field	Value
`SPDMOperation`	0x02

81 The length of the `SPDM Storage Pending Info` response data for this version of the *SPDM to Storage Binding Specification* shall be 12 bytes.

82 **Table 8 — SPDM Storage Pending Info response data**

Byte offset	Field	Size (bytes)	Description
0	DataLength	2	The `DataLength` field shall return the number of available bytes in the response data, not including any Pad fields. This value might exceed the Allocation Length in the `IF-RECV` command.
2	StorageBindingVersion	2	The `StorageBindingVersion` for devices that implement this version of the SPDM to Storage Binding Specification shall be set to `0x1000`. Table 7 — Storage binding version format defines the format of this field.

Byte offset	Field	Size (bytes)	Description
4	PendingInfoFlag	4	The `PendingInfoFlag` field shall contain flags regarding the `SPDM Storage Pending Info` response. • Bit[0]. ValidResponse. Shall be set to `1` to indicate that a pending response is available and the `ResponseLength` field contains a valid pending response length. This bit shall be set to `0` to indicate that a valid response is not available. All other values reserved.
8	ResponseLength	4	The `ResponseLength` field shall return the number of available bytes in the currently pending response message for the connection indicated in the `ConnectionID` field. The value in this field shall be set to a non-zero value if the `PendingInfoFlag`.`ValidResponse` bit is set to `1`. Otherwise, the value in this field shall be set to zero.

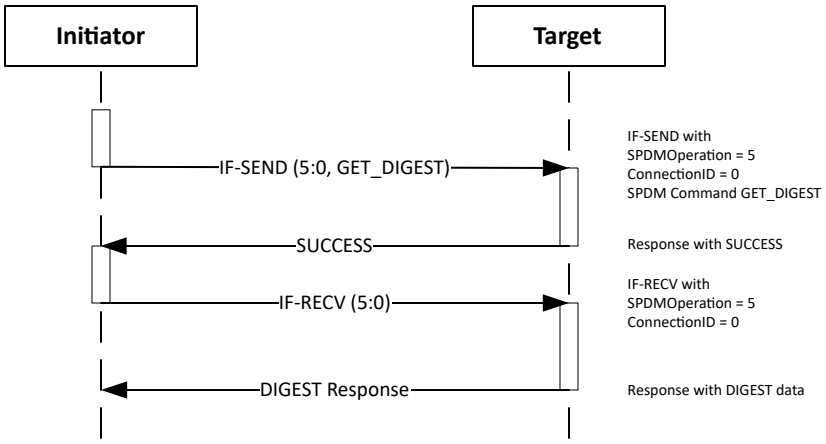
6.3 SPDM Storage Message

The `SPDM Storage Message` uses `IF-SEND` and `IF-RECV` to send and receive SPDM messages, as defined in DSP0274, between an initiator and a target. This binding specification requires that SPDM be used in a request/response flow, and does not support targets sending messages asynchronously.

As Figure 1 — Storage message flow shows, `SPDM Storage Message` commands are unidirectional. An individual command can only send data or receive data. To handle this restriction, the Storage Binding Specification uses the `IF-SEND` and `IF-RECV` commands in a pair. An `IF-SEND` is used to send a request and any associated data from the initiator to the target. The target completes this command with status. The initiator then sends an `IF-RECV` from the initiator to the target. Finally, the target sends the SPDM response message and status to complete the `IF-RECV`.

Figure 1 — Storage message flow

87



88

6.3.1 SPDM Storage Message IF-SEND

89 The `SPDM Storage Message` `IF-SEND` shall be used to send an SPDM request from an initiator to a target. The data buffer shall contain request data that is of length `Transfer Length`, including any Pad bytes.

90 The `SPDMOperation` field in the `IF-SEND` shall be set to `0x5` to indicate `SPDM Storage Message`.

91 The data buffer transmitted from the initiator to the target shall be an SPDM request.

92

6.3.2 SPDM Storage Message IF-RECV

93 The `SPDM Storage Message` `IF-RECV` shall be used to transfer an SPDM response from a target to an initiator. The response data shall be in a data buffer, the size of which is less than or equal to the value in the Allocation Length field.

94 The `SPDMOperation` field in the `IF-RECV` shall be set to `0x5` to indicate `SPDM Storage Message`. The value in the `ConnectionID` field in the `SPDM Storage Message` `IF-RECV` shall be the same as the value in the `ConnectionID` field in the corresponding `SPDM Storage Message` `IF-SEND`.

95 The data buffer transmitted from the target to the initiator shall be the SPDM response data, including any `ERROR` response data.

96

6.3.3 SPDM Storage Message status

97 Two status values apply to `SPDM Storage Message`s. The storage protocol defines errors related to the operation of the protocol. Errors in transmitting an `SPDM Storage Message` or associated data shall result in a protocol defined error response. Error responses related to an SPDM message are reported using the SPDM `ERROR` response. An SPDM `ERROR` response should be associated with a successful status for the storage protocol, unless a protocol condition occurs in conjunction with the SPDM `ERROR` response.

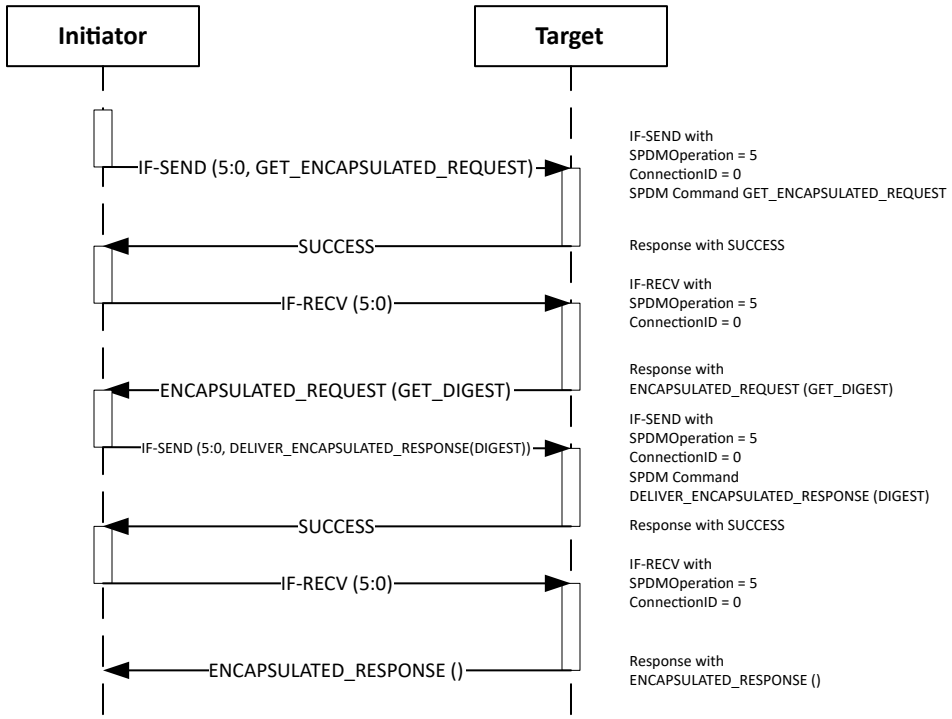
98 ### 6.3.4 Encapsulated request flow

99 Since the SPDM Storage Binding does not support asynchronous requests from a target to an initiator, any message flows that require a target to send a request to the initiator shall use the SPDM Encapsulated Request Flow.

100 Storage encapsulated message flow shows the SPDM encapsulated message flow for a storage protocol. When an initiator starts an encapsulated flow, the initiator shall use the same `ConnectionID` for all messages in the encapsulated flow.

101 **Figure 2 — Storage encapsulated message flow**

102



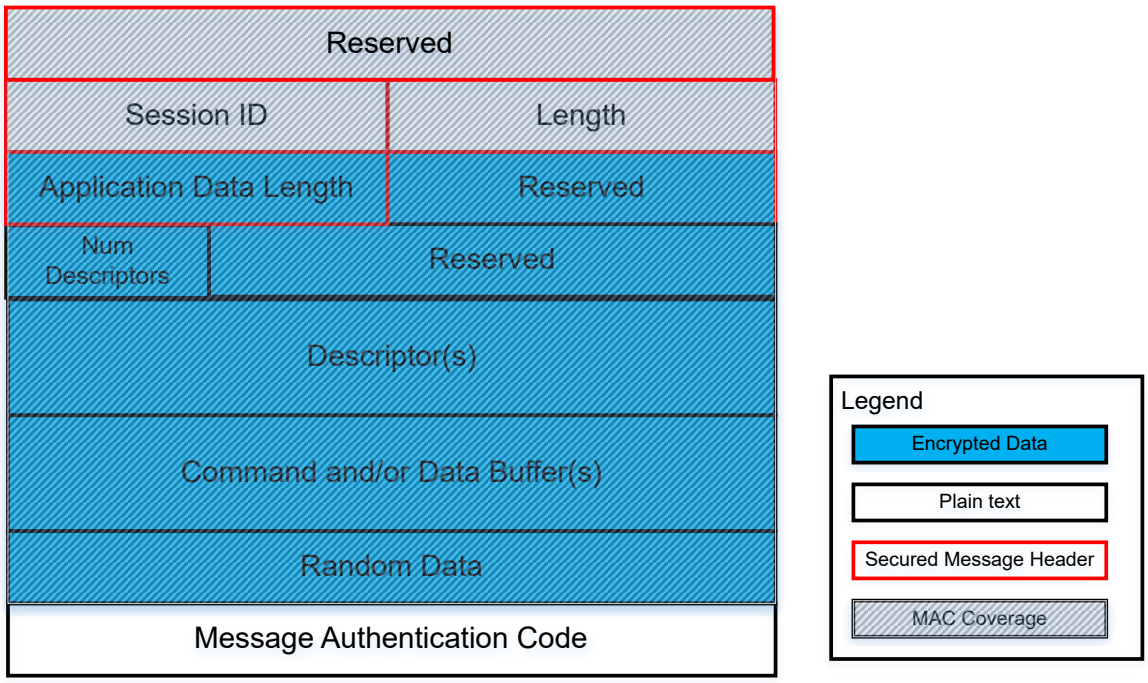
103 ## 6.4 SPDM Storage Secured Message

104 The `SPDM Storage Secured Message` uses `IF-SEND` and `IF-RECV` to send and receive Secured Messages, as defined in DSP0277, between an initiator and a target. The `SPDM Storage Secured Message` uses descriptors and buffers to describe the messages and/or data that is transported by the Secured Message. The data buffer for an `SPDM Storage Secured Message` shall follow the format shown in Figure 31131 — SPDM Storage Secured Message data buffer and described in this clause. When an initiator sends a Secured Message to a target, the initiator shall use `IF-SEND` with the buffer format described in this clause. When a target responds to an initiator, the target shall fill the buffer supplied by `IF-RECV` with data in the buffer format described in this clause.

105 If encryption is enabled, the message data is encrypted from `Application Data Length` through `Random Data`, inclusive. The Message Authentication Code calculation covers the fields from the `Reserved` at offset 0 through `Random Data`, inclusive. The fields from `Num Descriptors` through `Command or Data Buffer`, inclusive, shall be treated as the `Application Data` as described by DSP0277.

106 **Figure 31131 — SPDM Storage Secured Message data buffer**

107



108 Note: Figure 31131 — SPDM Storage Secured Message data buffer is not drawn to scale.

109 Table 9 — SPDM Storage Secured Message data buffer field descriptions describes the fields shown in Figure 31131 — SPDM Storage Secured Message data buffer.

110 **Table 9 — SPDM Storage Secured Message data buffer field descriptions**

Byte offset	Field	Size (bytes)	Description
0	Reserved	4	Reserved.
4	Session ID	4	Shall be the `Session ID` field as DSP0277 defines.
8	Length	2	Shall be the `Length` field as DSP0277 defines.
10	Application Data Length	2	Shall be the `Application Data Length` field as DSP0277 defines.
12	Reserved	4	Reserved.

Byte offset	Field	Size (bytes)	Description
16	Num Descriptors	1	Shall be the number of `SPDM Storage Secure Message Descriptor` elements in this SPDM Storage Secured Message data buffer.
17	Reserved	3	Reserved.
20	SPDM Storage Secure Message Descriptor	16 * `Num Descriptors`	Shall be `Num Descriptors` instances of the `SPDM Storage Secure Message Descriptor` .
20 + 16 * `Num Descriptors`	Secure Message Data	Variable	Shall be `Num Descriptors` buffers that contain data described by the `SPDM Storage Secure Message Descriptor` s. See Table 10 — SPDM Storage Secured Message descriptor format.
Variable	Command or Data Buffer	Variable	Shall be the command and/or data buffer described by the `SPDM Storage Secure Message Descriptor` s.
Variable	Random Data	Variable	Shall be the `Random Data` field as DSP0277 defines.
Variable	Message Authentication Code (MAC)	Variable	Shall be the `Message Authentication Code` field as DSP0277 defines.

111 Table 10 — SPDM Storage Secured Message descriptor format describes the `SPDM Storage Secured Message Descriptor` format.

112 **Table 10 — SPDM Storage Secured Message descriptor format**

Byte offset	Field	Size (bytes)	Description
0	Reserved	1	Reserved.
1	DescType	1	Shall be the type of the descriptor element. See Table 11 — SPDM Storage Secured Message descriptor types. One `SPDM Storage Secured Message` shall contain no more than one descriptor in the `Command` category (`DescType` = `0x01` to `0x04`) and shall optionally contain no more than on descriptor in the `Data Buffer` category (`DescType` = `0x40`). As SPDM does not use an associated data buffer, so a descriptor of type SPDM (`DescType` = `0x04`) shall not include an associated data buffer.
2	Status	1	In an type SPDM storage secured message request, this field shall be reserved. In an SPDM storage secured message response, this field shall be populated as Secured message encapsulated status defines.

Byte offset	Field	Size (bytes)	Description
3	Reserved	1	Reserved.
4	Length	4	Shall be the length of the corresponding `Secure Message Data` element.
8	Offset	4	Shall be the offset of the corresponding `Secure Message Data` element from the start of the `SPDM Storage Secured Message` data buffer. If the `Length` and/or `Offset` for one `Secure Message Data` element overlaps with any other `Secure Message Data` element in this message, the device shall return a protocol error. See SPDM Storage protocol status.
12	Reserved	4	Reserved.

113 Table 11 — SPDM Storage Secured Message descriptor types describes the details of the `DescType` field in the `SPDM Storage Secured Message Descriptor`.

114 **Table 11 — SPDM Storage Secured Message descriptor types**

`DescType` Value	Category	Description
0x01	Command	Shall describe an NVMe command as the NVM Express Base Specification describes.
0x02	Command	Shall describe a SCSI Command Descriptor Block as the SCSI Architectural Model describes.
0x03	Command	Shall describe an ATA command as the ATA/ATAPI Command Set describes.
0x04	Command	Shall describe an SPDM request or response message as DSP0274 describes. This `DescType` is used to transmit SPDM commands in a secured session.
0x40	Data Buffer	Shall describe a data transfer buffer. The data transfer buffer shall immediately follow the command with which it is associated.
All others	Reserved	All others reserved.

115 ### 6.4.1 Use of descriptors

116 The use of `SPDM Storage Secured Message Descriptor`s in the `SPDM Storage Secured Message` allow a single Secured Message to transmit all elements of a single command to an endpoint. A Requester shall send only one `SPDM Storage Secured Message` per connection at a time, and shall wait for a response or timeout before sending the next `SPDM Storage Secured Message` in that connection. The use of descriptors shall not allow endpoints to bypass any message restrictions outlined in SPDM (DSP0274) or Secured Messages (DSP0277).

117 If a device cannot process the provided command or data buffer, the device shall return an appropriate error defined by the protocol specification.

118 # 7 Storage protocol specific behavior

119 ## 7.1 Transcript hash calculation

120 Transcript hashes shall be calculated as DSP0274 describes. The transcript hash shall be calculated over the contents of `SPDM Storage Message` data buffers, but shall not include the fields in `IF-SEND` and `IF-RECV`. The transcript hashes shall not include any pad data from the data buffer.

121 # 8 Storage specific accommodations

122 ## 8.1 Device reset handling

123 Each storage protocol or its underlying transport defines one or more reset operations. This clause specifies SPDM behaviors related to these resets. Storage protocol and underlying transport resets that are not specified in this section shall not have an effect on the SPDM protocol behavior.

124 When using the SCSI protocol on a SAS transport, the mapping of resets to SPDM behavior shall follow the behavior that Table 12 — SAS transport reset to SPDM mapping defines.

125 **Table 12 — SAS transport reset to SPDM mapping**

SAS Reset Event	SPDM Behavior
LOGICAL UNIT RESET task management function	Device reset.
Link reset sequence with hard reset	Device reset.

126 When using the ACS protocol, the mapping of resets to SPDM behavior shall follow the behavior that Table 13 — ACS protocol reset to SPDM mapping defines.

127 **Table 13 — ACS protocol reset to SPDM mapping**

ACS Reset Event	SPDM Behavior
COMRESET with Software Settings Preservation disabled	Device reset.

128 When using the NVMe protocol, the mapping of resets to SPDM behavior shall follow the behavior that Table 14 — NVMe reset to SPDM mapping defines.

129 **Table 14 — NVMe reset to SPDM mapping**

NVMe Reset Event	SPDM Behavior
NVMe subsystem reset	Device reset.

130 ## 8.2 Status response hierarchy

131 When using the commands defined by this specification, error responses can occur in multiple domains, or at multiple layers of the interface between the Requester and Responder. The device shall report errors in the following domain order, and is not required to report status from a lower priority domain when a higher priority domain has a non-successful completion status. The following list gives the order in which error domains are processed, with the lowest numbered domain being given highest priority.

1. Transport status. Transport status shall encompass all status that indicate whether the request or

response was correctly received by the other device. Transport errors can include, but are not limited to, errors at the physical, link, or transport layer. These errors shall be reported using protocol or transport-defined error messages.

2. SPDM Storage protocol status. The domain of SPDM Storage protocol status shall define status reporting, as the SPDM Storage protocol status clause defines, for errors that are detected in the format or contents of an `IF-SEND` or `IF-RECV` that contains an SPDM Storage protocol command, as Table 4 — SPDM operation codes defines.
3. SPDM protocol status. SPDM protocol status shall report status as DSP0274 and DSP0277 define. Any hierarchy or prioritization of error status reporting between these specifications shall be as these specifications define.
4. Secured message encapsulated status. Status that is reported in response to a command sequence that occurs in an SPDM Storage Secured Message session shall be reported in a session using a Secured message encapsulated error. See Secured message encapsulated status.

132 ## 8.2.1 SPDM Storage protocol status

133 This specification refers to errors that fall into the SPDM Storage protocol status category as `SPDM Storage Protocol Error`s. Devices shall report `SPDM Storage Protocol Error`s using protocol specific error codes. A Responder shall send status using the same protocol as the protocol used for the associated request.

134 ### 8.2.1.1 SCSI protocol status

135 Table 15 — SCSI reporting of SPDM Storage Protocol Errors defines the reporting of `SPDM Storage Protocol Error`s using the SCSI protocol.

136 **Table 15 — SCSI reporting of SPDM Storage Protocol Errors**

SPDM Storage Protocol Error	SCSI Status	SCSI Sense Key	SCSI ASC/ASCQ	Comments
Success	GOOD	NO SENSE	NO ADDITIONAL SENSE INFORMATION	Normal command completion.
Invalid or unsupported value in Command management	CHECK CONDITION	ILLEGAL REQUEST	INVALID FIELD IN CDB	No data shall be transferred.
Invalid Transfer Length in IF-SEND	CHECK CONDITION	ILLEGAL REQUEST	INVALID FIELD IN CDB	No data shall be transferred.
Other Invalid Command Parameter	CHECK CONDITION	ILLEGAL REQUEST	INVALID FIELD IN CDB	No data shall be transferred.

137 ### 8.2.1.2 ATA protocol status

138 Table 16 — ACS reporting of SPDM Storage Protocol Errors without SDR defines the reporting of `SPDM Storage`

`Protocol Error`s using the ATA/ATAPI Command Set when Sense Data Reporting (SDR) is not available (due to SDR not being available, SDR being in a disabled state, or the `SENSE DATA AVAILABLE` bit is set to `0`).

139 **Table 16 — ACS reporting of SPDM Storage Protocol Errors without SDR**

SPDM Storage Protocol Error	ATA Status Field	ATA Error Field	Comments
Success	0x50	0x00	Normal command completion.
Invalid or unsupported value in Command management	0x51	0x04	No data shall be transferred.
Invalid Transfer Length in IF-SEND	0x51	0x04	No data shall be transferred.
Other Invalid Command Parameter	0x51	0x04	No data shall be transferred.

140 If the device supports SDR, SDR is enabled, and the `SENSE DATA AVAILABLE` bit is set to `1`, then status shall be reported as Table 15 — SCSI reporting of SPDM Storage Protocol Errors defines, with the addition that Bit[1] of the `ATA Status Field` shall be set to `1`.

141 8.2.1.3 NVMe protocol status

142 Table 17 — NVMe reporting of SPDM Storage Protocol Errors defines the reporting of `SPDM Storage Protocol Error`s using the NVMe protocol.

143 **Table 17 — NVMe reporting of SPDM Storage Protocol Errors**

SPDM Storage Protocol Error	NVMe Status Code Type	NVMe Status Code	NVMe Do Not Retry Bit	Comments
Success	Generic Command Status	Successful Completion	0	Normal command completion.
Invalid or unsupported value in Command management	Generic Command Status	Invalid Field in Command	1	No data shall be transferred.
Invalid Transfer Length in IF-SEND	Generic Command Status	Invalid Field in Command	1	No data shall be transferred.
Other Invalid Command Parameter	Generic Command Status	Invalid Field in Command	1	No data shall be transferred.

144 8.2.2 Secured message encapsulated status

145 The status of an encapsulated command in secured message is returned using an `SPDM Storage Secured Message Descriptor`.

146 If the command is an SPDM Request, then the status shall be returned in an `SPDM Message`. This response takes the form of either an SPDM formatted response to the request, or an `ERROR` response.

147 If the command is an NVMe, SCSI, or ATA command, the response status shall be returned in the `Status` field of a `Data Buffer` type `SPDM Storage Secured Message Descriptor`. See Table 10 — SPDM Storage Secured Message descriptor format. In this case, the `Status` field shall contain a `StatusCode` as Table 18 — Encapsulated Response Status Codes defines. If the `StatusCode` is any value other than `Success`, the Data Buffer pointed to by the `SPDM Storage Secured Message Descriptor` shall only contain `StorageErrorData`, if any is defined. If `StorageErrorData` is not defined for the given `StatusCode`, the `Length` and `Offset` fields of the `Data Buffer` type `SPDM Storage Secured Message Descriptor` shall be set to `0`.

148 **Table 18 — Encapsulated Response Status Codes**

StatusCode Value	StatusCode	StorageErrorData	Description
0x00	Success	None	The command completed without an error.
0x01	General Error	None	The device encountered an error while processing the request. Protocol specific mechanisms may contain additional information about the error.
0x02	Invalid Command	None	The command code is not recognized or not supported in an SPDM storage secured message.
0x03	Invalid Field	None	The command contains one or more fields that are not valid.
0xFF	Vendor Defined	See Vendor defined secured message encapsulated status	The format of the Vendor Defined StatusCode shall be as Vendor defined secured message encapsulated status defines.

149 **8.2.2.1 Vendor defined secured message encapsulated status**

150 The format for the `Vendor Defined Secured Message Encapsulated Status` shall be as Table 19 — Vendor defined secured message encapsulated status defines. This format follows the `SVH` format as defined in DSP0274.

151 **Table 19 — Vendor defined secured message encapsulated status**

Offset	Field	Length (bytes)	Description
0	ID	1	Shall be one of the values in the `ID` column of "Registry or standards body ID" table as defined in DSP0274.
1	VendorLen	1	Length in bytes of the `VendorID` field. If the definition of the data in `ErrorData` belongs to a standards body, this field shall be 0. Otherwise, the definition of the data in `ErrorData` belongs to the identified vendor and therefore, this field shall be the length indicated in the `Vendor ID` column of "Registry and standards body ID" table for the respective `ID` defined in DSP0274.
2	VendorID	VendorLen	If `VendorLen` is greater than zero, this field shall be the ID of the vendor corresponding to the `ID` field. Otherwise, this field shall be absent.
2 + `VendorLen`	ErrorDataLen	2	Shall be the length, in bytes, of the `ErrorData` field.
2 + `VendorLen` + `ErrorDataLen`	VenErrorData	ErrorDataLen	Shall contain the vendor or standards body defined error data.

8.3 Padded transactions

153 Certain storage protocols require or support specifying the size of `IF-SEND` and `IF-RECV` transactions in increments of either 4 bytes or 512 bytes. If a command specifies a buffer size that is larger than the message that is being transmitted, the buffer requires a pad at the end of the buffer. In such a case, the SPDM message shall start from Byte[0] of the message. The pad bytes shall fill from after the end of the SPDM message until the end of the data buffer. For `IF-SEND`, the Requester shall use 0x00 for the pad value and the Responder shall ignore the pad value. For `IF-RECV`, the Responder shall use 0x00 for the pad value and the Requester shall ignore the pad value. See Transcript hash calculation for implications of pad values on transcript hash calculations.

8.4 Multi-path handling

155 A Requester is expected to use protocol specific commands to discover the relationship between different ports on the same device. When a Responder supports multiple paths (multiple routes over the transport to the same endpoint), SPDM defined information in the payload that is presented by the Responder shall be the same for all paths unless the change in the information reflects the path differences. The connection between a given Requester and a specific Responder port shall be considered a unique SPDM connection. Because SPDM connections are unique per device port, SPDM sessions cannot be shared across device ports.

156 Response data, such as certificate chains and measurements, should be the same regardless of which port is used to retrieve the data. In some cases, a response may include port-specific information, in which case the information is required to be specific to the port through which it is retrieved.

157 # 9 ANNEX A (informative) change log

158 ## 9.1 Version 1.0.0 (in progress)

- Initial release

10 Bibliography

DMTF DSP4014, *DMTF Process for Working Bodies 2.6*, https://www.dmtf.org/sites/default/files/standards/documents/DSP4014_2.6.pdf