Document Identifier: DSP0277

Date: 2020-08-05

Version: 1.0.0a

Secured Messages using SPDM Specification

Information for Work-in-Progress version:

IMPORTANT: This document is not a standard. It does not necessarily reflect the views of the 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 This document is available for public review and comment until superseded.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: http://www.dmtf.org/standards/feedback

Supersedes: None

Document Class: Normative

Document Status: Work In Progress

Document Language: en-US

13 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.

14 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patent owner or claimant, and shall have no liability or responsibility for costs or losses incurred if a standard is withdrawn or modified after publication, and shall be indemnified and held harmless by any party implementing the standard from any and all claims of infringement by a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.

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

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

17 CONTENTS

18 # 1 Foreword

19 The Platform Management Components Intercommunications (PMCI) Working Group prepared the *Secured Messages using SPDM Specification* (DSP0277).

20 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.

21 ## 1.1 Acknowledgments

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

- Patrick Caporale — Lenovo
- Nigel Edwards — Hewlett Packard Enterprise
- Daniil Egranov — Arm Limited
- Philip Hawkes — Qualcomm Inc.
- Brett Henning — Broadcom Inc.
- Jeff Hilland — Hewlett Packard Enterprise
- Theo Koulouris — Hewlett Packard Enterprise
- Donald Matthews — Advanced Micro Devices, Inc.
- Edward Newman — Hewlett Packard Enterprise
- Eliel Louzoun — Intel Corporation
- Jim Panian — Qualcomm Inc.
- Scott Phuong — Cisco Systems, Inc.
- Viswanath Ponnuru — Dell Technologies
- Xiaoyu Ruan — Intel Corporation
- Bob Stevens — Dell Technologies
- Nitin Sarangdhar — DMTF

23 # 2 Introduction

24 Secured Messages using SPDM specification defines the methodology that various PMCI transports can use to communicate various application data securely by utilizing SPDM. Specifically, this specification defines the transport requirements for SPDM records, which form the basis of encryption and message authentication.

25 Furthermore, this specification contains guidance and certain decisions that it defers to the binding specification which binds Secured Messages to a specific transport. Thus, the binding specification is expected to finalize those decisions or guidance by way of normalization or recommendation. The present specification was written with PMCI transports in mind, but nothing precludes specifying bindings to other transports.

26 ## 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.

27 # 3 Scope

28 This document defines a generic record format used to encrypt and authenticate any application data within SPDM's secure session. Also, relating to encryption, message authentication, and secure sessions, this specification further defines those areas in SPDM that the specification states are the responsibilities of the transport layer.

29 ## 3.1 Normative references

30 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.

- DMTF DSP0274, *Security Protocol and Data Model (SPDM) Base Specification 1.1.0*, https://www.dmtf.org/sites/default/files/standards/documents/DSP0274_1.1.0.pdf
- *ISO/IEC Directives, Part 2, Principles and rules for the structure and drafting of ISO and IEC documents*, https://isotc.iso.org/livelink/livelink.exe?func=ll&objId=4230456&objAction=browse&sort=subtype
- IETF RFC5234, *Augmented BNF for Syntax Specifications: ABNF*, January 2008, https://tools.ietf.org/html/rfc5234

31 ## 3.2 Terms and definitions

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

33 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.

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

35 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.

36 The terms that DSP0274 define also apply to this document.

3.3 Symbols and abbreviated terms

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

4 Secured Message

Starting with SPDM 1.1, SPDM describes at a very high and abstract level a construct, called a record, to encrypt and authenticate data within a session. SPDM places the responsibility of the details and definition of the record onto the transport layer. The manifestation of this record in this specification is called a Secured Message.

A Secured Message shall only be used within a secure session. Specifically, a Secured Message can be used in any phase of a secure session, such as the session handshake phase and the application phase.

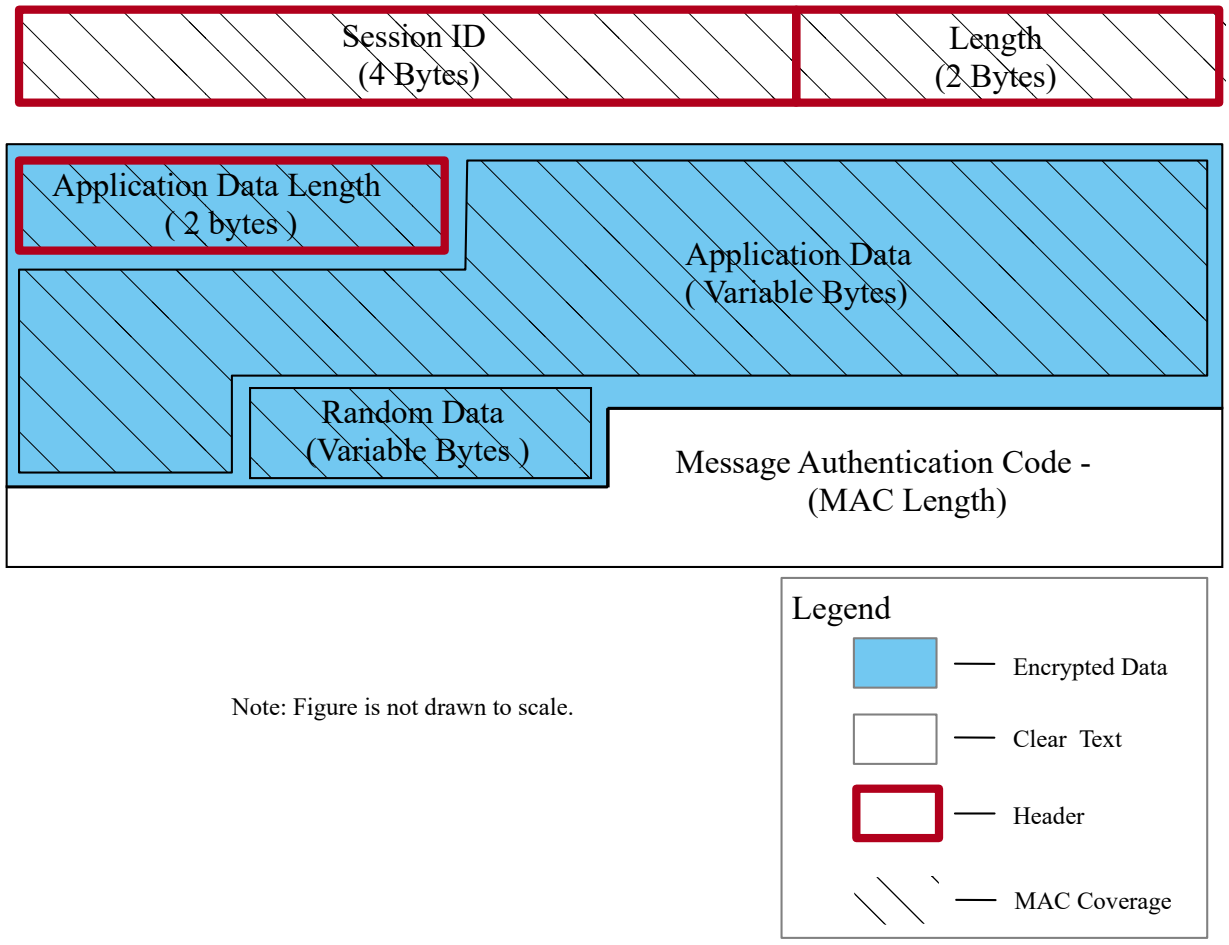
To support a Secured Message, an SPDM endpoint shall support one or both of message authentication and encryption. Additionally, an SPDM endpoint shall support one or more AEAD algorithms as defined in the SPDM specification. Finally, an SPDM Responder shall select an AEAD algorithm according to SPDM specification (DSP0274).

4.1 Secured Message format

The Secured Message format figure illustrates the format used to encrypt or authenticate application data. Because this specification creates a single format for all application data, application data shall also include those SPDM messages that are required to be sent within a session such as `KEY_UPDATE` messages.

Secured Message format

46



Note: Figure is not drawn to scale.

47 The Secured Message fields definition table defines the exact format.

48 **Secured Message fields definition**

Offset	Field	Length (bytes)	Description
0	Session ID	4	The Responder and Requester can use this field to bind all session information such as secrets and keys. This field shall be the same value as the `SessionID` field in the Session-Secrets-Exchange response. To ensure forward and backward compatibility, this field shall never change and shall always be first.
4	Length	2	This field shall indicate the remaining length of data in Secured Messages.
6	Application Data Length	2	This field shall be the length of application data.

Offset	Field	Length (bytes)	Description
8	Application Data	P	This field shall contain either application data or in-session SPDM messages.
8 + P	Random Data	R	This field should contain random data of random length.
See Description	Message Authentication Code (MAC)	Variable	This field is for illustrative purposes only. The actual location and details of the MAC in the cipher text shall adhere to the AEAD specification for the selected AEAD cipher in `ALGORITHMS` response. AEAD algorithms usually append the MAC to the end of the cipher text.

49 Except for the MAC and Application Data, all fields are in little endian.

50 The purpose of the random data field is to further obfuscate the application data by hiding the real length of the application data. This prevents information from being derived from the real length of application data. In certain scenarios that do not require this obfuscation, the `Random Data` field can have a length of zero.

51 The `Length` field shall be the sum of the length of the following fields:

- Application Data Length (if present)
- Random Data (if present)
- Application Data
- MAC

52 The Field presence requirement table describes the presence requirement for each field in Secured Messages. Initially, both the Requester and Responder advertise their capabilities through `GET_CAPABILITIES` and `CAPABILITIES` messages. The two capabilities of interest are encryption and message authentication. For a session to provide encryption, both the Requester and Responder need to support encryption. Likewise, for a session to provide message authentication, both the Requester and Responder need to support message authentication. Lastly, for a session to provide both encryption and message authentication, both the Responder and Requester have to support both.

53 The most common denominator of capabilities between an SPDM Requester and its Responder shall determine the capabilities for all sessions between them. SPDM does not allow a Requester and Responder to support different capabilities for each individual session. In other words, if both the Requester and Responder support both message authentication and encryption, all their sessions shall support message authentication and encryption; not one or the other but both. Likewise, if there are no common capabilities between the two, Secured Messages will not work.

54 If a session can only provide message authentication, the `Message Authenticaton Only` column is applicable. If a session provides both, the `Encryption and Message Authentication` column is provided. Only one column applies per session. An encryption only session shall be prohibited. If no columns apply, this specification is not applicable.

55 In the applicable column, a value of **Present** shall indicate the respective field is present in the Secured Message. A value of **Absent** shall indicate the respective field shall not be present in Secured Messages.

56 **Field presence requirement**

Field	Message Authentication Only	Encryption and Message Authentication
Length	Present	Present
Session ID	Present	Present
Application Data Length	Absent	Present
Random Data	Absent	Present
Application Data	Present	Present
MAC	Present	Present

57 For some transport bindings, the Application Data field will need a specified format to ensure correct processing at the receiver. For example, if the Application Data field can carry messages from a range of protocols, then the Application Data field might need a field indicating which protocol to use for processing the message in the Application Data field. If required, such formatting shall be specified by the binding specification.

58 ## 4.2 Secured Message protection

59 Secured Messages utilize Authenticated Encryption with Associated Data (AEAD) cipher algorithms in much the same way that TLS 1.3 does. See the SPDM specification (DSP0274) for an overview of AEAD algorithms.

60 ### 4.2.1 AEAD encryption keys and other secrets

61 SPDM's key schedule produces four major secrets that are used at certain points in the session and each secret is bound to a particular direction of transmission. The encryption keys and initialization vector (IV) are derived from these four major secrets. See Key Schedule in SPDM specification (DSP0274) for more details.

62 ### 4.2.2 AEAD requirements

63 This clause discusses the requirements for each parameter to the AEAD functions depending on the capabilities of the session. See the Application data in SPDM specification (DSP0274) for the AEAD functions and more details. The references below shall be interpreted according to DSP0274 definitions.

64 In general, the MAC covers the associated data and the plain text. Specifically, the MAC covers all fields in Secured Message. The default length of the MAC shall be 16 bytes for AES-GCM and ChaCha20-Poly1305. The transport binding can specify a different MAC length.

65 #### 4.2.2.1 Message Authentication Only session

66 For sessions that are capable of only supporting message authentication, the associated data, `associated_data`, for AEAD shall be the concatenation of the following fields in this order:

1. Session ID
2. Length
3. Application Data

67 The text to encrypt, `plaintext`, for AEAD shall be null. Consequently, the text to decrypt, `ciphertext`, shall also be null.

68 #### 4.2.2.2 Encryption and Message Authentication session

69 The associated data, `associated_data`, for AEAD shall be the concatenation of the following fields in this order:

1. Session ID
2. Length

70 The text to encrypt, `plaintext`, for AEAD shall be the concatenation of these fields in this order:

1. Application Data Length
2. Application Data
3. Random Data

71 The text to decrypt, `ciphertext`, shall be the encrypted portion of the Secured Message and the MAC.

72 ### 4.2.3 Per-message nonce derivation

73 The nonce shall never be transmitted in Secured Messages. This means that both the Responder and Requester must internally track the nonce. To ensure proper tracking, the Requester and Responder shall follow the nonce derivation schedule laid henceforth.

74 Before the creation of the first Secured Message in the session for a given major secret and its derived encryption and IV keys, both the Responder and Requester shall start with a 64-bit sequence number with a value of zero. For each record, both SPDM endpoints shall follow these steps as prescribed:

1. Zero extend the sequence number to `iv_length` according to the selected AEAD cipher suite in `ALGORITHMS` messages.
2. Perform a bitwise XOR of the zero-extended sequence number with the appropriate IV derived in the SPDM key schedule.
   - The output of this step is called the per-message nonce.

3. Increment the sequence number by a value of one for the next Secured Message.

75 Because different secrets are used for different directions of data transmission, each endpoint would have to track two sequence numbers: one for the reception and the other for the transmission.

76 Lastly, when a `KEY_UPDATE` occurs, the sequence number shall reset to 0 before sending the first Secured Message using the new session keys.

77 #### 4.2.3.1 Other per-message nonce requirements

78 A Secured Message shall not reuse a sequence number. Furthermore, an SPDM endpoint shall not send Secured Messages out of sequence. The Per-message nonce derivation clauses describes the proper sequence.

79 ### 4.2.4 Encryption requirements

80 A single Secured Message shall contain the complete cipher text as produced by a single invocation of `AEAD_Encrypt` using the appropriate encryption key for the given direction of transmission, the appropriate per-message nonce, and the selected AEAD Cipher Suite in `ALGORITHMS`. No two or more Secured Messages shall use the same nonce.

5 Compatibility

This specification is decoupled from the SPDM specification to avoid unnecessary updates here whenever SPDM changes. If a tighter coupling to the SPDM specification is desired, the transport binding specification should describe it.

6 Version support

To advertise the supported Secured Message version of an SPDM Requester for a particular transport, the Secured Message version format table defines the fields necessary to specify the transport binding specification version of Secured Messages.

Secured Message version format

Bit	Field	Value
`[15:12]`	`MajorVersion`	Shall be the version of the transport binding of the Secured Message using SPDM specification with changes that are incompatible with one or more functions in earlier major versions of the specification.
`[11:8]`	`MinorVersion`	Shall be the Version of the transport binding of the Secured Message using SPDM specification with changes that are compatible with functions in earlier minor versions of this major version specification.
`[7:4]`	`UpdateVersionNumber`	Shall be the Version of the transport binding of the Secured Message using SPDM specification with editorial updates but no functionality additions or changes. Informational; possible errata fixes. Ignore when checking versions for interoperability.
`[3:0]`	`Alpha`	Shall be the Version of the transport binding of the Secured Message using SPDM specification with editorial updates but no functionality additions or changes. Informational; possible errata fixes. Ignore when checking versions for interoperability.

The transport binding specification of Secured Messages may offer a more descriptive or definitive statement on compatibility between major, minor and update versions.

The Secured Message version list format table describes a format to list all supported versions of Secured Messages for a given transport.

Secured Message version list format

Offset	Field	Size (in byte)	Description
0	`VersionCount`	1	Shall indicate the total number (T) of Secured Message versions listed in `VersionsList`
`1+`	`VersionsList`	T * 2	Shall list all versions that are supported by an SPDM Requester for the given transport. The format for this field shall be the format described by the Secured Message version format table.

6.1 Version selection

Version discovery and selection occurs during Session-Secrets-Exchange. First, the SPDM Requester shall advertise its

list of supported version through the `OpaqueData` field of a Session-Secrets-Exchange request. The Requester shall use the Secured Message opaque data format to specify its list in Supported version list data format.

91 Lastly, the Responder shall select a version among the ones that is supported by the Requester and communicate the selected version in the `OpaqueData` field in the Session-Secrets-Exchange response. The Responder shall use the Secured Message opaque data format to specify its selected version in Version selection data format. From that point on, both the SPDM Requester and Responder shall not change this version for that session. If the Responder cannot select a version, an ERROR response shall be sent with `ErrorCode=InvalidRequest`.

92 # 7 Transport requirements or allowances

93 This clause and subclauses describe various requirements or flexibility allowed at the transport layer.

94 ## 7.1 Transmission reliability

95 Secured Messages rely on the transport to perform reliable lossless delivery. The transport defines the mechanisms to ensure the transmission of data which can include retries. Furthermore, this specification expects the transport to either deliver Secured Messages in order or allow the receiver to determine the correct order of transmission of Secured Messages. In an event that transmission or reception fails, an SPDM Requester or Responder may terminate the session or restart a new one.

96 ## 7.2 Certain SPDM message allowances

97 If possible, the transport binding specification should take full advantage of asynchronous and bidirectional communication to allow messages such as `KEY_UPDATE` and `HEARTBEAT` to be sent directly from a Responder without any other assistance such as a sideband alerting mechanism or SPDM's `GET_ENCAPSULATED_REQUEST` mechanism. The transport binding specification shall address this.

98 ## 7.3 ERROR response message allowances

99 Furthermore, the `ERROR` message may be sent without an SPDM request when the error code is a decryption error (`ErrorCode=DecryptError`) to indicate that the Secured Message that was received could not be decrypted properly. In addition, both the SPDM Requester and Responder may send an `ERROR` message with `ErrorCode=DecryptError` . This is especially useful for data sent at the application layer. In other words, in this scenario, the `ERROR` response message is behaving as a response to the inability to decrypt or authenticate the received Secured Message. In the event an SPDM endpoint receives this particular error message, the SPDM endpoint should terminate the session.

100

8 Secured Messages opaque data format

101 In many SPDM requests and response, an opaque data field exist to accommodate transport, standard organizations or vendor specific use cases. The Secured Message general opaque data table defines the general format for all opaque data fields in SPDM. All opaque data fields in SPDM messages shall utilize the format defined by Secured Message general opaque data. This format allows an SPDM message to contain multiple vendor or standard bodies opaque data without collision.

102 **Secured Message general opaque data table**

Offset	Field	Length (bytes)	Description
0	SpecID	4	Shall be 0x444D546. This value is the hexadecimal representation of the string DMTF. The purpose of this field is to help distinguish opaque data defined by this specification from other opaque data.
4	OpaqueVersion	1	This field shall identify the format of the remaining bytes. This value shall be 1.
5	TotalElements	1	Shall be the total number of elements in `OpaqueList`.
6	Reserved	2	Reserved
8+	OpaqueList	Variable	Shall be a list of Opaque Elements.

103 The Opaque element table defines the format for each element in `OpaqueList`.

104 **Opaque element table**

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 the SPDM Specification (DSP0274).
1	VendorLen	1	Length in bytes of the `VendorID` field. If the data in `OpaqueElementData` belongs to a standards body, this field shall be 0. Otherwise, the data in `OpaqueElementData` belongs to the 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	OpaqueElementDataLen	2	Shall be the length of `OpaqueElementData`.

Offset	Field	Length (bytes)	Description
X : 4 + VendorLen	OpaqueElementData	Variable	Shall be the data defined by the vendor or standards body.
Y : X + 1	AlignPadding	1, 2 or 3	If `X` does not fall on a 4-byte boundary, this field shall be present and of the correct length to ensure `Y` ends on a 4-byte boundary. This field shall be all zeros.

105 ## 8.1 Secured Message opaque element data format

106 The Secured Message opaque element data format implements the Opaque element for use cases specific to this specification. The Secured Message opaque element table describes the implementation.

107 **Secured Message opaque element table**

Offset	Field	Length (bytes)	Description
0	ID	1	Shall be zero to indicate DMTF.
1	VendorLen	1	Shall be zero. Note: DMTF does not have a vendor registry.
2	OpaqueElementDataLen	2	Shall be the length of the remaining bytes excluding the `AlignPadding`.
4	SMDataVersion	1	Shall identify the format of the remaining bytes. The value shall be one.
5	SMDataID	1	Shall be the identifier for the Secured Message data type. Later sections of this specification describes the allowed values.
X : 6	SMData	Variable	Shall be the data corresponding to `SMDataID`. Later sections of this specification describes this format.
Y : X + 1	AlignPadding	1, 2 or 3	See `AlignPadding` in Opaque element.

108 ### 8.1.1 Version selection data format

109 The Version selection data format table implements the Secured Message opaque element to communicate the selected Secured Message version. This data type shall only be allowed in a Session-Secrets-Exchange Response. An SPDM Responder populates this information.

110 **Secured Message version selection data format table**

Offset	Field	Length (bytes)	Description
0		5	See equivalent bytes in Secured Message opaque element table.

Offset	Field	Length (bytes)	Description
5	SMDataID	1	Shall be a value of zero to indicate Secured Message version selection.
6	SelectedVersion	2	Shall be the selected Secured Message Version. See Secured Message Version Format for the format of this field.

111 ### 8.1.2 Supported version list data format

112 The Supported version list data format table implements the Secured Message opaque element to list all the supported Secured Message versions. This data type shall only be allowed in a Session-Secrets-Exchange Request. An SPDM Requester populates this information.

113 **Supported version list data format**

Offset	Field	Size (in bytes)	Description
0		5	See equivalent bytes in Secured Message opaque element table.
5	SMDataID	1	Shall be a value of one to indicate Supported version list.
6	`SecuredMsgVers`	(T * 2) + 1	Shall be the format described in Secured Message version list format.

9 ANNEX A (informative)

9.1 Change log

Version	Date	Description
1.0.0a	2020-08-05	

116 # 10 Bibliography

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