Document Identifier: DSP1000

Date: 2024-04-15

Version: 1.0.0

Work Group Version: <Doc WG Version>

CPU TEE Secured Messages using SPDM Specification

Document Class: Normative

Document Status: Draft

Document Language: en-US

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.

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patent owner or claimant, and shall have no liability or responsibility for costs or losses incurred if a standard is withdrawn or modified after publication, and shall be indemnified and held harmless by any party implementing the standard from any and all claims of infringement by a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

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.

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

CONTENTS

1 Foreword

The CPU TEE Secured Messages using SPDM Specification (DSP1000) was prepared by the <DMTF Editing Body>.

DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems management and interoperability. For information about the DMTF, see http://www.dmtf.org.

2 Acknowledgments

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

- <first name and last name> – <company name>
-
-

<List the editor or editors for the current release first,followed by each contributor's name and company arranged in alphabetical order by contributors' last names.>

3 Introduction

This document defines specifications for implementing secure message exchange using SPDM for CPU TEE instances, facilitating secure intercommunication between different types of TEE instances.

3.1 Document conventions

Typographical 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.

4 Scope

This document defines specifications for implementing secure message exchange using SPDM for CPU TEE instances, detailing the integration of CPU TEE remote attestation processes with SPDM, including CPU TEE's X.509 certificate design and the design of CPU TEE's Measurements message content.

4.1 Normative references

The following referenced documents are indispensable for the application of this document. 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, version 1.1 or later,* https://www.dmtf.org/dsp/DSP0274

DMTF DSP0277, *Secured Messages using SPDM Specification 1.1.0,* https://www.dmtf.org/dsp/DSP0277

IETF RFC9334, *Remote ATtestation procedureS (RATS) Architecture, January 2023,* https://www.rfc-editor.org/rfc/rfc9334.html

IETF RFC5280, *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, May 2008,* https://www.rfc-editor.org/rfc/rfc5280.html

IETF RFC8949, *Concise Binary Object Representation (CBOR), December 2020,* https://www.rfc-editor.org/rfc/rfc8949.html

IETF RFC6920, *Naming Things with Hashes, April 2013,* https://www.rfc-editor.org/rfc/rfc6920.html

4.2 Terms and definitions

In this document, some terms have a specific meaning beyond the normal English meaning. Those terms are defined in this clause.

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.

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

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.

The terms defined in DSP0274 and DSP0277 apply to this document. The following additional terms are used in this document.

Term	Definition
Trusted Execution Environment (TEE)	A secure, isolated computing environment that protects sensitive operations from external threats, even from the host system.

Term	Definition
Concise Binary Object Representation (CBOR)	A binary data format that is more space-efficient than JSON, designed for use in systems where bandwidth and storage are at a premium.
Evidence	Data used to validate the authenticity or integrity of a digital process or claim, often involving cryptographic mechanisms.
Endorsements	Statements from trusted entities that affirm the credibility of a claim or identity, essential for establishing trust in secure systems.
Claims	Assertions about an entity's identity, attributes, or privileges, used in security contexts for access control and identity verification.
Attester	An entity that provides assurance about the state or integrity of a system, often through cryptographic means.
Verifier	An entity responsible for confirming the authenticity and validity of claims or evidence, critical in maintaining system security.
Software Guard Extensions (SGX)	A technology in Intel CPUs that creates secure enclaves for applications, protecting them from external software and even the operating system.

4.3 Symbols and abbreviated terms

The abbreviations defined in DSP0274 and DSP0277 apply to this document.

5 CPU TEE Secure Message

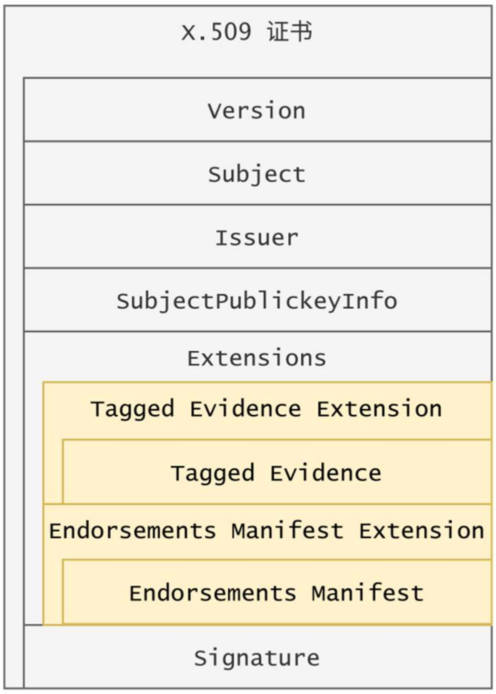
This specification describes the secure message exchange process between programs within CPU TEE instances and external programs. CPU TEE instances are one of the parties involved in secure message exchange. This specification utilizes the SPDM message exchange method described in the Security Protocol and Data Model (SPDM) Base Specification (DSP0274) and the SPDM secure message encoding format defined in the Secured Messages using SPDM Specification (DSP0277). By customizing the GET_CERTIFICATE/CERTIFICATE and GET_MEASUREMENTS/MEASUREMENTS messages, the remote attestation process of TEE and the message protocol of SPDM are combined, achieving secure message exchange.

To achieve this goal, it is necessary to define the X.509 certificate of CPU TEE and the Measurements Block of CPU TEE.

5.1 CPU TEE X.509 Cert

As one of the parties in SPDM message exchange (typically the Responder), each CPU TEE instance should have its own X.509 certificate chain. Unlike the three certificate models described in DSP0274, the X.509 certificate of CPU TEE is a special self-signed leaf certificate. The format definition part of this X.509 certificate refers to the Interoperable RA-TLS X.509 Cert and Evidence Formats draft. It remains an X.509 certificate compliant with IETF RFC5280, but its Extensions field contains a Tagged Evidence Extension and an optional Endorsements Manifest Extension.

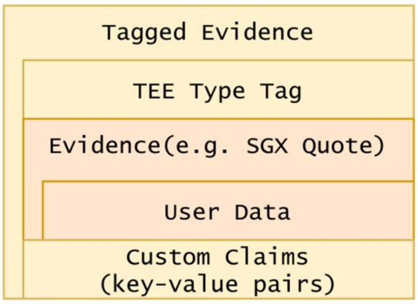
Figure 1 - CPU TEE X.509 Cert data format



5.1.1 Tagged Evidence Extension

The Tagged Evidence Extension is an X.509 certificate extension with OID 2.23.133.5.4.9.

Figure 2 - Tagged Evidence data format



The value of this extension, evidence-ext-value, is binary data serialized in CBOR format, as shown in Figure 2 - Tagged Evidence data format. Specifically, it is a fixed-length array with two elements, each tagged with CBOR Tag. Its definition is as follows:

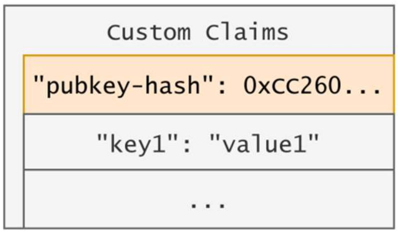
evidence-ext-value = tee-type-tag([evidence-data, custom-claims-data])

tee-type-tag is a CBOR Tag used to distinguish different TEE types or different types of evidence data.

evidence-data is of type byte string, representing the binary evidence data of the CPU TEE instance. The internal format of this field is specific to each type of TEE and is therefore opaque in this specification.

custom-claims-data is also of type byte string, representing another binary data serialized in CBOR format. Specifically, it is a Map type that stores user-defined Claims. These Claims are provided by programs running in the TEE instance, thus distinct from the TEE instance's own Claims.

Figure 3 - Custom Claims data format



The format of custom-claims-data is as shown in Figure 3 - Custom Claims data format, with its specific definition as follows:

custom-claims-data = { key : value, ... }

Where key is the identifier of the Claim, with a data type of string, typically named with a specific meaning. value is the value of the Claim, with a data type of byte string. This specification does not impose any limitations on the number of entries in custom-claims-data but does not allow the same key to appear more than once.

Although the key-value pairs stored in custom-claims-data can be provided by programs in specific scenarios, there must be one key with the value "pubkey-hash" in order for the CPU TEE X.509 certificate defined in this specification. Specifically, the value pubkey-hash-value records the hash value of the public key of the CPU TEE's X.509 certificate defined in this specification. Specifically, the structure of pubkey-hash-value is defined as follows:

pubkey-hash-value = [hash-alg-id, hash-value]

Where pubkey-hash-value is a two-element fixed-length array containing hash-alg-id and hash-value. hash-alg-id is an unsigned integer used to identify the hash algorithm used when calculating the public key hash value, with values referenced from the definitions in RFC6920. hash-value has a data type of byte string and represents the hash value of the SubjectPublicKeyInfo data structure of the CPU TEE's X.509 certificate. The hash value is calculated as follows:

hash-value = hash(SubjectPublicKeyInfo)

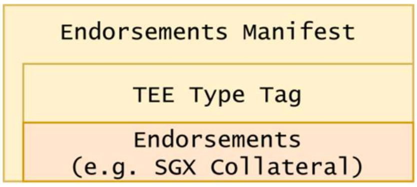
Where hash() represents a hash algorithm that should correspond to the hash algorithm type recorded in hash-alg-id.

5.1.2 Endorsements Manifest Extension

The Endorsements Manifest Extension stores additional data needed to verify evidence, typically including additional certificate chain data and X.509 CRL manifests. This extension is optional because the Verifier of remote attestation may have its own means of obtaining this Endorsements information. However, in some cases, the Verifier may need the Attester to pass on the Endorsements due to efficiency requirements or being in a restricted network.

The value of the Endorsements Manifest Extension is binary data serialized in CBOR format, termed endorsements-manifest-ext-value. Its data format is as shown below:

Figure 4 - Endorsements Manifest data format



Specifically, endorsements-manifest-ext-value is a byte string type with a CBOR Tag, and its specific definition is as follows:

endorsements-manifest-ext-value=tee-type-tag(endorsements-data)

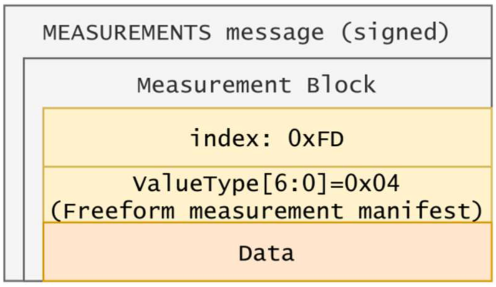
tee-type-tag is a CBOR Tag, and its value must be consistent with the tee-type-tag value in evidence-ext-value. endorsements-data is of type byte string, representing the binary endorsements data of the CPU TEE instance. The internal format of this field is specific to each type of TEE and is therefore opaque in this specification.

5.2 CPU TEE Measurements Definition

CPU TEE Measurements should include a description of the trust status information of the current TEE instance. According to DSP0277, the SPDM Requester can obtain the measurement information of the SPDM Responder by sending a GET_MEASUREMENTS message. This specification still uses this message to exchange measurement information between the Requester and Responder.

As a Responder, the TEE instance should provide at least one Measurement Block of type Measurement manifest with Index 0xFD, as shown in Figure 5 - CPU TEE Measurement Block data format, with DMTFSpecMeasurementValueType[6:0] = 0x04, indicating Freeform measurement manifest.

Figure 5 - CPU TEE Measurement Block data format


MEASUREMENTS message (signed)
Measurement Block
index: 0xFD
ValueType[6:0]=0x04
(Freeform measurement manifest)
Data

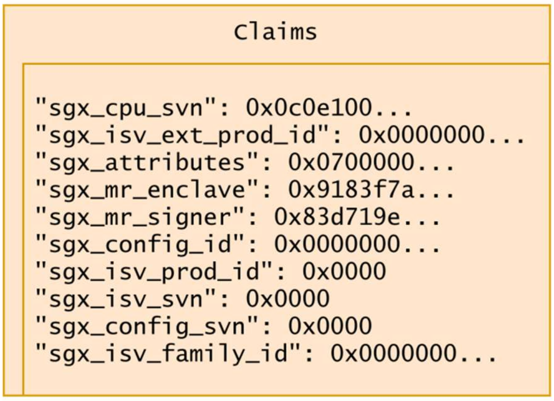
The Data field stored in this Measurement Block is binary data serialized in CBOR format, defined as follows:

measurement-data = { key : value, ... }

Where measurement-data is a Map type data, and the key-value pairs are each parsed from the Evidence of the current TEE instance. As the Evidence generated by different TEE types varies, the Claims are also slightly different, but this specification does not impose restrictions on them.

Figure 6 - Intel SGX Measurement Block Claims data format is an example of the Claims generated in an Intel SGX instance.

Figure 6 - Intel SGX Measurement Block Claims data format


Claims
"sgx_cpu_svn": 0x0c0e100...
"sgx_isv_ext_prod_id": 0x0000000...
"sgx_attributes": 0x0700000...
"sgx_mr_enclave": 0x9183f7a...
"sgx_mr_signer": 0x83d719e...
"sgx_config_id": 0x0000000...
"sgx_isv_prod_id": 0x0000
"sgx_isv_svn": 0x0000
"sgx_config_svn": 0x0000
"sgx_isv_family_id": 0x0000000...

6 ANNEX A (informative) change log

6.1 Version 1.0.0(2024-04-15)

- Initial release

7 Bibliography

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

Interoperable RA-TLS X.509 Cert and Evidence Formats, https://github.com/CCC-Attestation/interoperable-ra-tls/blob/main/README.md