



SPDM 1.3

New Features (Work-in-Progress)

Last Updated: August 31, 2022



Disclaimer

- The information in this presentation represents a snapshot of work in progress within the DMTF.
- This information is subject to change without notice. The standard specifications remain the normative reference for all information.
- For additional information, see the DMTF website.
- This information is a summary of the information that will appear in the specifications. See the specifications for further details.



Overview

- Asynchronous Notification
- Multiple Asymmetric Key
- Measurement Manifest
- Extendable Measurements
- Miscellaneous Features



ASYNCHRONOUS NOTIFICATION



Use Case and Requirements

- Use Case:
 - Receive changes to measurement data without polling.
 - Receive changes to certificate without polling
- Requirements:
 - Must work across multiple transports (including alliance partners).
 - Allow for non-polling mechanism (like an event)
 - Allow for other SPDM data changes (not just measurement data)
- Notes:
 - Specifically, to DMTF PMCI, PLDM has an eventing mechanism.
 - Not all Responders or Requester implement PLDM.
 - Also is not transport-agnostic.



Out of Scope

- Posted Events (events without acknowledgements) are pushed out of scope for the initial release of SPDM eventing. Future versions of SPDM can incorporate posted events if deemed necessary.
- Sending an SPDM event using PLDM event mechanism is out of scope of this slide set.





Events

- Allows for asynchronous notification to an event recipient over a secure session.
- Event recipient can subscribe or unsubscribe to event groups.
- New Request / Response:
 - GET_SUPPORTED_EVENTS/ SUPPORTED_EVENTS
 - Retrieves a list of supported event types for each supported event group.
 - SUBSCRIBE_EVENTS/ SUBSCRIBE_EVENTS_ACK
 - Subscribe or unsubscribe to certain event types in certain event groups.
 - SEND_EVENT / EVENT_ACK
 - Sends one or more events when they occur.
 - NOTE: An event notifier has to consider error handling, timing constraints and other complexities.
- GET_CAPABILITIES and CAPABILITIES:
 - `EVENT_CAP` bit to indicate that the SPDM endpoint supports eventing as an event notifier.



Events – Event Groups

- Event Groups are a collection of events defined by a standard organization or a vendor.
 - If a standards body is not listed in Table 54 — Standard body or vendor-defined header (SVH) of SPDM 1.2, please reach out to DMTF to be listed.
- SPDM DMTF Event Groups:
 - Event Lost
 - Indicates if the event notifier lost/discard/drop any events for any reason.
 - Measurement Update Event
 - Indicates the measurement index that has changed.
 - Certificate Event
 - Indicates if a certificate chain has changed.
 - Measurement Pre-Update Event
 - Indicates if one or more measurement indexes that will change due specifically to a firmware activation/update before the firmware activation will occur.

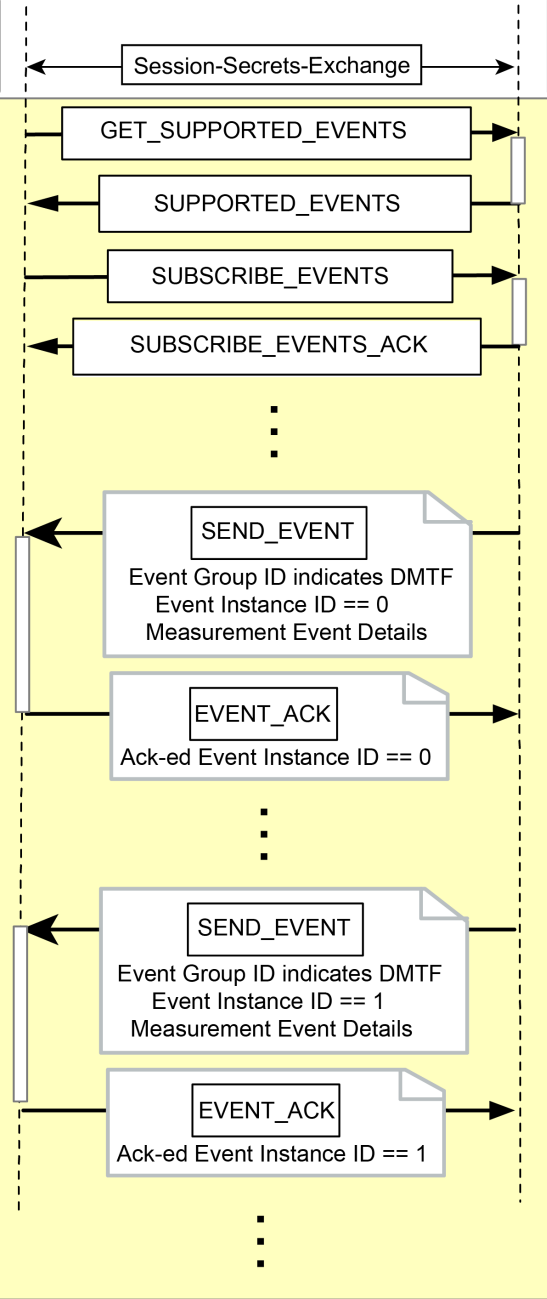



Events – Secure Sessions

- Events are bound to a sessions because sessions naturally offer:
 - Replay protection
 - Obfuscation
 - Message Authentication
- Replay protection and message authentication are the most crucial protections.
- Design Consideration
 - It is possible to standardize the event mechanism to work outside of a session, but this will complicate the event mechanism to offer equal protections as a session does. A possible alternative is to accept a reduce security posture.
 - No use case has been made for events to occur outside of a session.

Event Recipient

Event Notifier



Legend:
 Secure Session

Events - Flow



MULTIPLE ASYMMETRIC KEY PAIR



Background

- Various alliance partners made requests to allow for more than a single key-pair per asymmetric algorithm.
- Alias Certificate Model:
 - Changes to Layer 0 would void the ECA(s) of the chain in an alias model and thus would need to be re-created. Thus, in order to recreate the chain, an SPDm requester would need to re-establish device provenance before accepting the new layer 0.
- TPM uses a different key for Attestation than device identity.
- Different keys would allow different operations or be used by different bodies.



Background

- DMTF:
 - More than one key pair per algorithm can allow for different use cases assigned to them.
 - Can potentially improve the security posture because if one key pair is compromised the other key pairs associated with other operations of the SPDM endpoint may not be compromised.



Proposed Solution – Editorial Changes (of existing text)

- Remove single key-pair restriction per algorithm type in “Certificate and certificate chain”.
- For Slot 0, a device key-pair is required (i.e., shall). Either Device Cert model or Alias Cert model shall be supported in slot 0.
- In “Runtime authentication” section, every where “leaf certificate of Responder” is used, add a qualifier like “leaf certificate associated with the desired slot id”.
- For alias model, the device CA text needs to be normalized for slot 0. The other slots can have any chain they want and don’t need a Device CA in the chain or a Device Leaf Cert.



Proposed Solution – Editorials

- Introduce “Generic Certificate Model” which is just a chain of certificates with no requirements placed (by SPDM) on the root or intermediate CA certificates.
 - Either Device cert model or Alias cert model are required for slot 0. Generic Certificate Model is prohibited for Slot 0.
 - Device and Alias cert models have requirements placed by SPDM which would be “overreaching” for slot 1 – 7 as they can interfere with the data center’s CA policies and existing infrastructure.
 - Cert Slots 1 to 7 can be any of the 3 models.
- Note: Provisioning of a key pair is outside the scope of this specification.



Proposed Solution – Just a Clarification

- This proposal does not modify the SPDm message flows for authentication or measurements.



Proposed Solution – GET_CAPABILITIES and CAPABILITIES

- MULTI_KEY_CAP bit:
 - If set, the device is capable of supporting more than one key pair for one or more asymmetric algorithms.



Proposed Solution – Provisioning a Shared Key using the CSR flow

Example: Generate a certificate for slot 5 but use the same key pair as in slot 2. Slot 2 is using a key pair with key pair ID = 40.

1. Issue a GET_CSR using key ID = 40.
2. Get the CSR signed by desired CA authority.
3. Issue a SET_CERTIFICATE using key ID = 40 with slot ID = 5 and the signed certificate and chain.



Proposed Solution – Provisioning a cert chain and key from an external agent.

Example: Secure environment wants to provision slot 7 and 4 with the same key. The device is not capable of CSR support. Additionally, the device currently has KeyPairID = 23 unprovisioned.

1. Secure environment generates a key pair for the targeted device.
2. Secure environment generates two cert chains using the same key pair.
3. Secure environment issues SET_CERTIFICATE with slot ID = 7 and KeyPairID = 23 with one of the cert chain and provides key pair information as well.
4. Secure environment issues SET_CERTIFICATE with slot ID = 4 and KeyPairID = 23 with the other cert chain and provides the same key pair information as well.

Note: Method of providing Key Pair information is out of scope.



Key Association, Unassociation and Configuration

- New Requests and Responses will be added to do the following:
 - Discover all key pairs and their current configuration.
 - Associate or unassociate a key pair to/from a certificate slot.
 - Assign a specific usage (i.e., KeyUsageVector) to a key pair.
 - Number of new requests and responses are TBD.
 - Format of new requests and responses are TBD.



EXTENDABLE MEASUREMENT



Background

- An SPDM Responder may use the MEASUREMENTS response to report to the Requester a set of measurements with Responder's signature on **concatenation of this set of measurements**.
- Some Responders manage measurements in a different manner using an Extend Register (ER) + Measurement Extension Log (MEL).
 - Measurements are extended / accumulated to ER.
 - MEL saves the concatenation of this set of measurements.
 - ER guarantees integrity of MEL.
 - Responder does not sign MEL. Responder signs only ER.
- Responder must report MEL and signed ER to Requester.
- **Problem statement:** SPDM 1.2 does not support Responder sending MEL and signed ER to Requester.



ER Calculation

- Responder uses an ER to accumulate measurement blocks in HW.
- At boot, initialize ER = a constant string
- Extend(ER, MeasurementBlock) function:
 $ER := \text{hash}(ER \parallel \text{MeasurementBlock})$
- Responder must keep all measurement blocks that have been extended to ER in Measurement Extension Log, which must match ER. MEL does not need to be signed.
- Responder signs the ER (as part of GET_MEASUREMENTS).
- MEL+ ER + signature on ER are transmitted to Requester.



SPDM Support Proposal

- Capability: Allocate a bit in Responder CAP “SUPPORT_EXTEND”.
- Measurement Extension Log:
 - New message (GET_MEASUREMENT_EXTENSION_LOG) with chunking
- ER and signature:
 - Available from MEASUREMENTS response
 - Issue “Add new DMTFSpecMeasurementValueType definition for accumulative digest (i.e. hash extend)”
<https://github.com/DMTF/Security-TF/issues/1903>



MEASUREMENT MANIFEST



Background

Measurement manifest as defined in SPDm:

The measurement manifest of DMTFSpecMeasurementValueType refers to a manifest that describes contents of other indexes.

Another potential case could be (not defined in SPDm yet):

The measurement manifest itself contains measurements as well as descriptive information.

Relevant issues:

- Reference Measurement (<https://github.com/DMTF/Security-TF/issues/560>)
- Provide a way to figure out which measurement blocks are in TCB (<https://github.com/DMTF/Security-TF/issues/1368>)



Terminology

Align the term in TCG specification (see reference section)

- **Reference Manifest**

Describe the expected/baseline/golden value.

Example: SWID, CoSWID, CoMID

- **Evidence Manifest**

Describe the actual value.

Example: TPM PCR/EventLog, DICE TcbInfo.

NOTE:

An evidence manifest MAY provide info about endorsement or reference manifest location.



Purposes of SPDM Measurement Manifest

1. It is an **evidence manifest**. It provides metadata to help a Requestor/verifier make sense of what is reported at each measurement index and any attributes that come with that value (e.g., is it part of the TCB, ownership info, etc.)
2. Optionally, It contains the identification or location information on “**reference manifest**”.
 - Support both a single “global” reference for all indices, and each index having its own reference.
 - If the reference is simple enough it could be encapsulated in the measurement manifest itself (but must be signed by someone other than the Responder). Or the measurement manifest may contain a pointer to the reference.

This slide deck uses term *RMM* (Reference Measurement Manifest) to refer to the data structure that conveys reference measurement values.



MISCELLANEOUS FEATURES



Miscellaneous Features

- **Certificate Erase Functionality**
 - Using SET_CERTIFICATE, requester can erase a certificate chain in slots 1 to 7 inclusively.
- **Get SPDM Endpoint Info**
 - New request and response to retrieve additional information such as UUID.
- **Raw Bitstream in Measurement Indices**
 - Changes to GET_MEASUREMENTS to allow Requester to retrieve raw bitstreams greater than 64KB.