SPDM 1.3 Features

May 2023

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.

Alliance Partners and Adopters

SPDM's Overall Goals

- All SPDM features fall into at least one of these main goals:
  - Device Attestation and Authentication
  - Secure Communication over any transport
- Device Attestation and Authentication
  - The ability to attest various aspects of a device such as firmware integrity and device identity
- Secure Communication over any Transport
  - Provide the ability to secure communication of any data or management traffic over any transport
  - Work with industry partners to ensure data in-flight is secure for all parts of the infrastructure (e.g. storage, network fabrics, etc.)

SPDM Feature Summary

- Version 1.0:
  - Measurement Support
  - Device Attestation and Authentication
- Version 1.1:
  - Secure Session
    - Public Key Exchange
    - Symmetric Key Exchange
  - Mutual Authentication
- Version 1.2:
  - Supports installation of certificates
  - Allows for alias certificates derived from device certificates
  - Send and receive large SPDM messages (chunks)
  - Added SM2, SM3, SM4 algorithms to supported list
  - New OIDs added
  - Deprecated basic mutual authentication in CHALLENGE and CHALLENGE_AUTH

SPDM 1.3 Feature Additions

- Added Eventing Mechanism
  - Allows either side of SPDM communication to notify another side about any changes in its state during secure session.
- Multiple Keys support
  - Added Generic Certificate Model.
  - Added Multiple Asymmetric Key capability.
- Measurement Enhancements
  - Added ability to retrieve only measurements that are pending changes but not applied yet
  - Added ability for Responder to notify Requester if *certain measurement was changed after it was reported to Re*quester in case when measurements are requested one-by-one in multiple GET_MEASUREMENT requests.
  - Measurement Extension Log (MEL) and Hash Extended Measurement (HEM) for tracking changes including changes to measurements
  - Add a standard manifest format for a measurement block
- Miscellaneous:
  - Added GET_ENDPOINT_INFO for collecting generic data about endpoint participating in SPDM communication

SPDM 1.3 Feature Deprecation

- Session
  - Removed restriction on Session ID reuse.

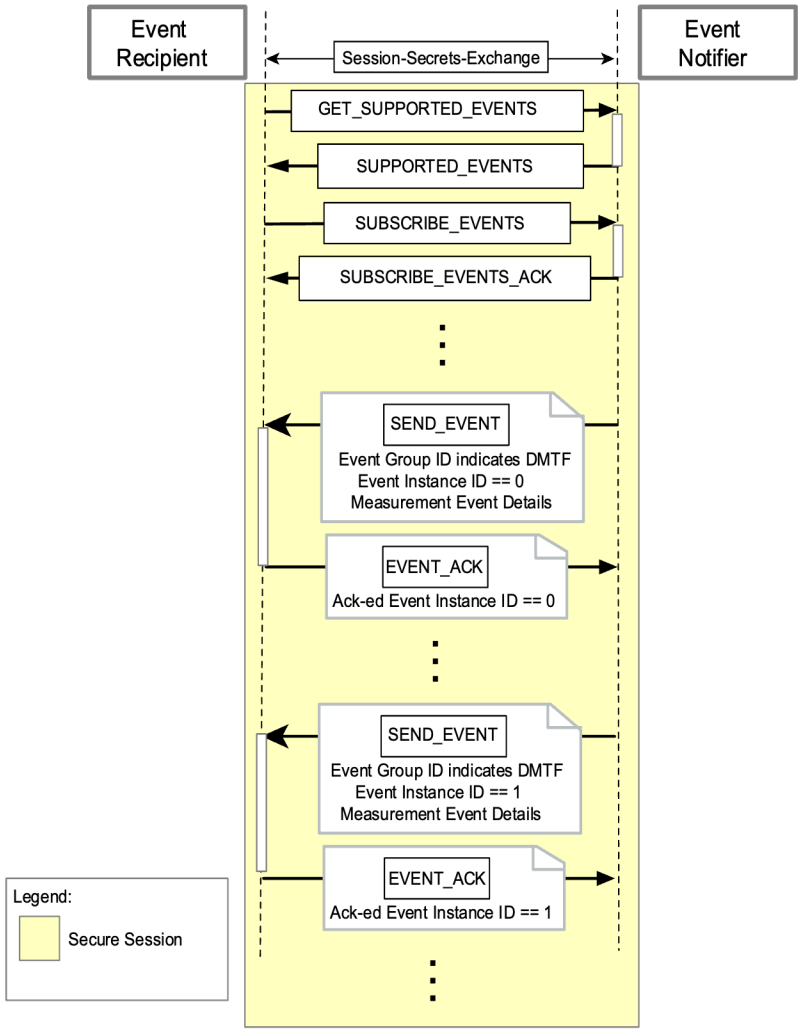
SPDM Change Awareness

- Statement of Backwards Compatibility:
  - SPDM message format will maintain bit-wise and semantic compatibility for existing fields.
    - SPDM may append new fields to an existing message.
    - SPDM may make use of reserved values.
    - SPDM may deprecate a valid value.
  - SPDM may make operational changes to fix a security issue or strengthen the security posture of the operation even if they are technically incompatible.

- Therefore, SPDM 1.3 contains changes that may be deemed technically incompatible with prior versions.
  - Please see change notes at the end of DSP0274 1.3 for details.
  - The Version field is in every packet and can be used to identify differences.

Event Mechanism

- All event notifications happen inside a Secure Session
- Event Recipient can collect information about Event Types supported, and then subscribe to interesting ones.
- Event Types could be extended by other standards bodies



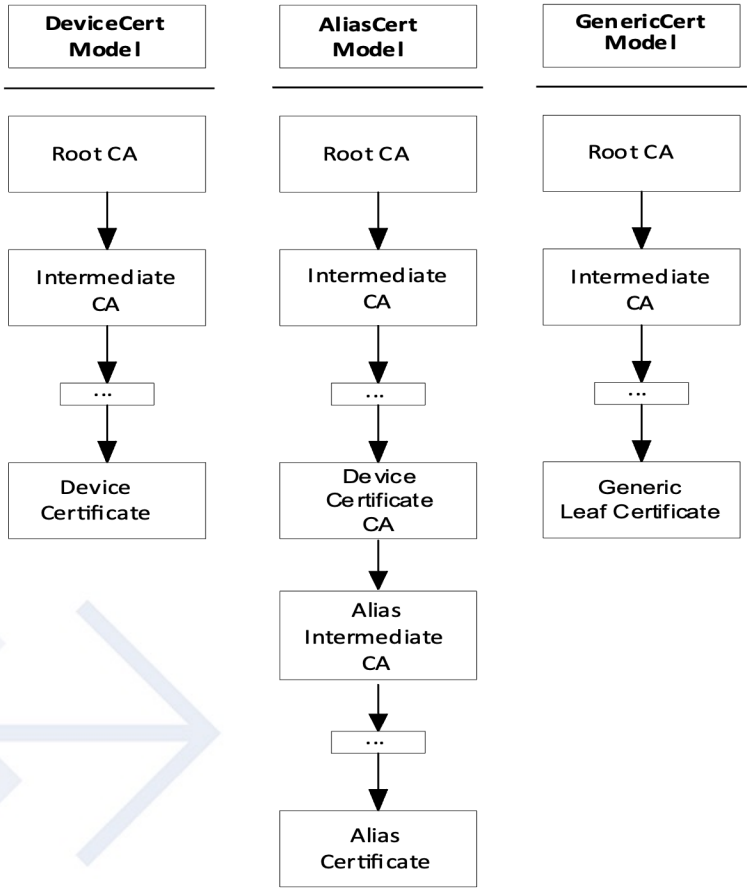
Multi Key Support

- Previous versions of SPDM only allowed one key pair per negotiated asymmetric algorithm
- Ability to use more than one key pair for a negotiated asymmetric algorithm
  - Up to 8 key pairs supported per asymmetric algorithm
- Every key pair could be dedicated for use case, like different key pairs for CHALLENGE and GET_MEASUREMENTS signature generations
- Requester is allowed to associate each key pair with an individual device certificate to enable one or more use cases
- Key pairs are identified by a unique KeyPairID

Generic Certificates Support

- What is a Generic Certificate or Certificate chain?
  - A Certificate or Certificate Chain that could not be qualified as a Device Certificate nor Alias Certificate.
- New Feature
  - Generic Certificate model is introduced to support Multiple Asymmetric Keys use cases.
  - Generic Certificate Model is the most flexible (or least restrictive) of the certificate models.
  - Generic Certificate Model applies to certificates in slots greater than 0.
    - A Device or Alias Certificate is required in slot 0.

SPDM Certificate Models



New Measurements

- `NewMeasurementRequested` field is introduced in the request attributes of the GET_MEASUREMENTS request.
  - If Responder has any changes affecting measurements that are requested by Requester but not yet applied (for example, pending changes due to a firmware update), then these new measurement values should be returned instead of current measurements (if requested using the value in the field above)
  - If there are no pending changes, then current measurements are returned regardless of the value in `NewMeasurementRequested` field

Measurement Extension Log (MEL) and Hash-Extended Measurements (HEM)

- Responder may support reporting of measurements thru an “extend” scheme
  - Initialize HEM = HashSize bytes of 0s
  - For each extend operation, perform HEM = hash(Concatenate(HEM, DataToExtend)) for all data elements to extend
- The MEL is the collection of DataToExtend
  - Could include configuration measurements, firmware measurements, version number, etc.
- An example of such a scheme is the Platform Configuration Register "extend" function in Trusted Platform Modules.
- There is a new MeasurementValueType 0x08 introduced for HEM

Structured Manifest format for a measurement block

- Data structure that describes the contents of other indices or contains measurements itself.
- Either Free Format or Structured
  - Free Format is implementation specific
  - Structured Format provides a Standards body or vendor-defined header, and manifest data in the format defined by the Standards body or vendor

Endpoint Info

- The GET_ENDPOINT_INFO request message retrieves general information from an endpoint.
  - The SubCode parameter is used to differentiate between operations.
  - The message supports a signature.
- Currently only one Subcode is defined: ***DeviceClassIdentifier***
  - The **DeviceClassIdentifier** response returns information that can be used to identify the class of device for the Responder in question.
    - For instance, DeviceClassIdentifier could contain PCI Vendor ID and Device ID fields.