SPDM 1.4 Features

June 2025

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

CXL Compute Express Link™

HDBaseT™

mipi® alliance

OPEN Compute Project

PCI SIG

TRUSTED® COMPUTING GROUP

nvm EXPRESS®

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.)
- Support latest cryptography standards
  - Especially post quantum crypto (PQC) algorithms, such as ML-DSA, ML-KEM, and SLH-DSA.

SPDM 1.0 – SPDM 1.3 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
- Version 1.3:
  - Eventing mechanism
  - Multiple key support
  - Measurement enhancements (new measurements and measurement extension log)
  - Structured manifest format
  - ENDPOINT_INFO messages

SPDM 1.4 Feature Additions

- PQC support in algorithm negotiation, certificates, and key pair information messages
  - ML-KEM (FIPS 203), ML-DSA (FIPS 204), SLH-DSA (FIPS 205)
- Certificate slot management
  - Defines “banks” of certificate slots, for managing certificates of different signing algorithms.
  - A bank contains up to 8 slots of certificates of the same signing algorithm.
  - A device may support multiple signing algorithms, hence multiple banks.
- Miscellaneous:
  - Added SET_KEY_PAIR_RESET_CAP
  - Added salt length requirement for RSA-PSS schemes.
  - Minor clarifications and fixes

Post Quantum Algorithm Support

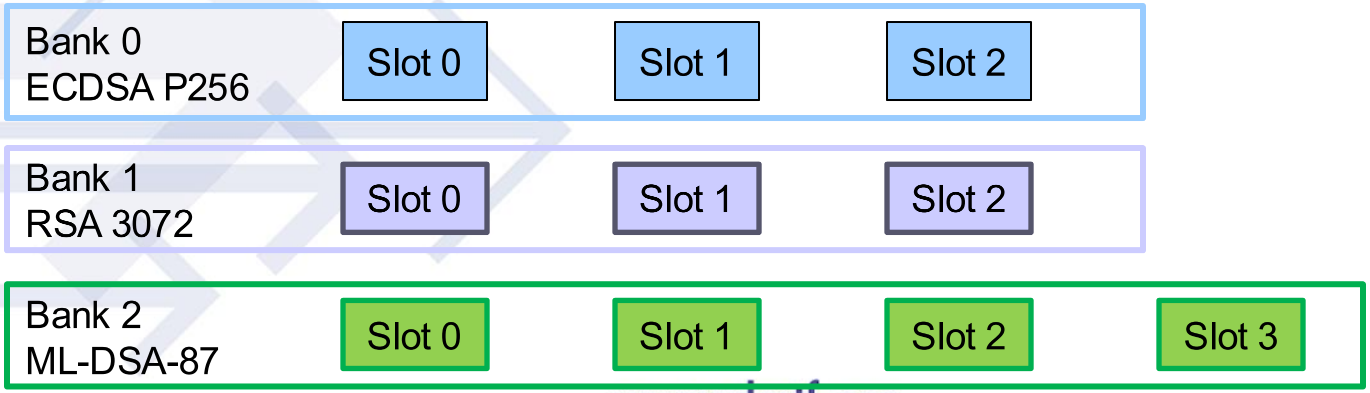
- Supports following standards
  - FIPS 203 ML-KEM for key encapsulation, all security categories
  - FIPS 204 ML-DSA for digital signature, all security categories
  - FIPS 205 SLH-DSA for digital signature, all parameters sets
  - NIST SP 800-227 Recommendations for Key-Encapsulation Mechanisms
- No hybrid support in SPDM 1.4.
  - NEGOTIATE_ALGORITHMS/ALGORITHMS cannot negotiate 2 or more algorithms (e.g., a traditional algorithm and a PQC algorithm) for a given category.

SPDM Messages Impacted by PQC

Message	Added / expanded fields for PQC	Digital Signature – added ML-DSA and SLH-DSA	Key Exchange – added ML-KEM
(“core” messages)			
CERTIFICATES	Yes		
CHALLENGE_AUTH		Yes	
ENDPOINT_INFO		Yes	
MEASUREMENTS		Yes	
KEY_EXCHANGE / _RSP	Yes	Yes	Yes
FINISH		Yes	
CSR		Yes	
SLOT_MANAGEMENT / _RESP	Yes		
(“supporting” messages)			
NEGOTIATE_ / ALGORITHMS	Yes		
SET_CERTIFICATE	Yes		
SET_KEY_PAIR_INFO	Yes		
KEY_PAIR_INFO	Yes		
(other messages)			

Certificate Slot Management With Bank

- Today, many devices support only one signing algorithm
  - The device can have up to 8 slots (numbered 0-7) that store certificate chains associated with the supported signing algorithm.
- When multiple signing algorithms are supported by the device, each algorithm can support up to 8 slots.
  - The bank of slots is implicitly selected via NEGOTIATE_ALGORITHMS / ALGORITHMS, providing no way to manage other banks of certificates.
- SPDM 1.4 provides commands to manage certificate chains
  - Every signing algorithm is allocated a bank.
  - The selected signing algorithm does not affect the new certificate slot management commands.



Bank 0
ECDSA P256
Slot 0 Slot 1 Slot 2

Bank 1
RSA 3072
Slot 0 Slot 1 Slot 2

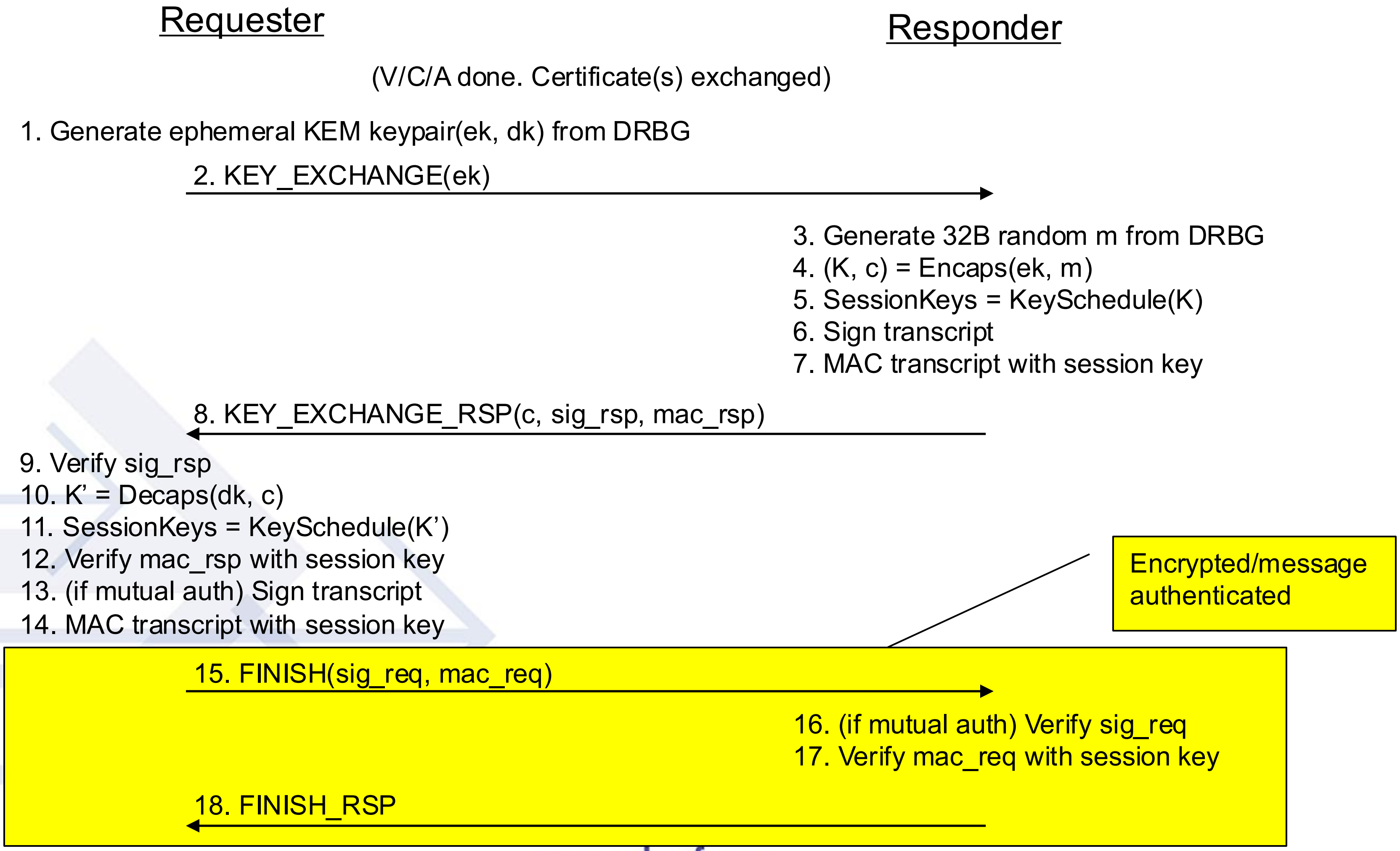
Bank 2
ML-DSA-87
Slot 0 Slot 1 Slot 2 Slot 3

Backup

References

- SPDM 1.4.0 https://www.dmtf.org/sites/default/files/standards/documents/DSP0274_1.4.0.pdf
- FIPS 203 https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.203.pdf
- FIPS 204 https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.204.pdf
- FIPS 205 https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.205.pdf
- SP 800-227 Initial Public Draft https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-227.ipd.pdf

KEY_EXCHANGE with ML-KEM
(HANDSHAKE_IN_THE_CLEAR == FALSE)



Requester

Responder

(V/C/A done. Certificate(s) exchanged)

1. Generate ephemeral KEM keypair(ek, dk) from DRBG

2. KEY_EXCHANGE(ek)

3. Generate 32B random m from DRBG
4. (K, c) = Encaps(ek, m)
5. SessionKeys = KeySchedule(K)
6. Sign transcript
7. MAC transcript with session key

8. KEY_EXCHANGE_RSP(c, sig_rsp, mac_rsp)

9. Verify sig_rsp
10. K' = Decaps(dk, c)
11. SessionKeys = KeySchedule(K')
12. Verify mac_rsp with session key
13. (if mutual auth) Sign transcript
14. MAC transcript with session key

Encrypted/message authenticated

15. FINISH(sig_req, mac_req)

16. (if mutual auth) Verify sig_req
17. Verify mac_req with session key

18. FINISH_RSP