Guy Fedorkow
Juniper Networks
gfedorkow@juniper.net
Kevin Micciche
HPE Aruba Networking
Kevin.micciche@hpe.com
Paul Bottorff
HPE Aruba Networking
Paul.bottorff@hpe.com
Maik Seewald
Cisco Systems
maseewald@cisco.com

July 24, 2025

Proposed Module-Lattice-Based Digital Signature Amendment to 802.1AR-2018

This document outlines a proposed amendment to IEEE 802.1AR-2018 to add support for Post Quantum Cryptography (PQC).

For the most part, this change is simply to add the Object Identifiers (OIDs) for soon-to-be-standardized Module-Lattice-Based Digital Signature (ML-DSA, FIPS-204, https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.204.pdf) in the ITU-T X.509 certificate format specified by IETF (https://datatracker.ietf.org/doc/draft-ietf-lamps-dilithium-certificates/).

The proposal is simply to add ML-DSA / Dilithium (and not subtracting anything). We are not expecting to add support for other NIST PQC algorithms such as SPHINCS+ / SLH-DSA or LMS/XMSS.

In recommending quantum-safe algorithms, NIST has selected several new algorithms. This proposal focuses on ML-DSA as the best match for the device identity application:

- ML-DSA key sizes (and hence certificates) are larger than classical RSA and ECC, but not unreasonably so, and computational requirements seem tractable for small devices like Trusted Platform Modules
- NIST has identified ML-DSA-87 as the only algorithm accepted in Commercial National Security Algorithm Suite 2.0 (CNSA2.0, https://media.defense.gov/2025/May/30/2003728741/-1/-1/0/CSA_CNSA_2.0_ALGORITHMS.PDF)
- The SPHINCS+ alternative has been approved as FIPS-205, but requires substantially larger keys, making it unattractive for small devices like TPMs

- The stateful signing algorithm LMS/XMSS can issue only a limited number of signatures until its state space is exhausted, and keys must be replaced, rendering it unattractive for long-lived applications like DevID.

ML-DSA is defined in three security strengths; we would recommend adding all three security strengths – ML-DSA-44, ML-DSA-65, ML-DSA-87 to 802.1AR.

It may be desirable to add an informational note to the standard indicating that government agencies intend to require ML-DSA-87, in spite of the large key sizes. [see CNSA2.0, and note https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/evaluation-criteria/security-(evaluation-criteria, to be added to the next revision of SP 800-57 part 1]

We note that this proposed specification change is dependent on final approval of IETF *draft-ietf-lamps-dilithium-certificates*.

Proposed Changes to the 802.1AR Specification

IEEE Std 802.1AR-2018 mentions support for RSA and ECC in passing in several paragraphs. ML-DSA should be added to these lists.

The technical details for algorithm support are given in Section 9, now covering RSA and two variants of ECC. This contribution contains a proposed new Section 9.4 to cover ML-DSA.

The conformance clause and PICS Proforma Annex A should be amended to include new sections to cover ML-DSA.

Any required additions to the management MIB should also be included in the amendment.

IEEE Std 802.1AR-2018 also references the Trusted Computing Group documents for provisioning DevID in TPM1.2. These sections should be updated to cross-reference TPM2.0 as well. A footnote to the existing TPM1.2 cross reference might note that TPM1.2 is not defined to support ML-DSA.

Proposed new 802.1AR Section

9.4 ML-DSA

9.4.1 Algorithms and parameters

Module-Lattice Digital Signature Algorithm (ML-DSA) signature schemes are defined in NIST FIPS 204, and are defined at three security strengths (listed in increasing strength): ML-DSA-44 (128-bit), ML-DSA-65 (192-bit), and ML-DSA-87 (256-bit). The algorithms are fully defined by the identifiers in FIPS 204, and require no distinguishing parameters. Use of the algorithms in X.509 certificates is specified in draft-ietf-lamps-dilithium-certificates (an IETF work-in-progress). HashML-DSA variants (FIPS 204, s 5.4) are not to be used, but ExternalMu-ML-DSA (https://datatracker.ietf.org/doc/draft-ietf-lamps-dilithium-certificates/, Appendix D) is permitted.

Compliance to NIST CNSA2.0 (https://media.defense.gov/2025/May/30/2003728741/-1/-1/0/CSA_CNSA_2.0_ALGORITHMS.PDF) requires ML-DSA-87.

9.4.2 Key generation

An RNG used by a DevID module to generate keys for this signature suite shall have sufficient entropy to generate keys with different security strengths for each variant, as shown in Table X.

Algorithm	RNG Strength
ML-DSA-44	At least 128 bits
ML-DSA-65	At least 192 bits
ML-DSA-87	At least 256 bits

Entropy requirements are specified by FIPS-204, Section 3.6.1.

9.4.3 signatureAlgorithm

The `signatureAlgorithm` field (8.8) value conforms to the general ASN.1 structure specified by RFC 5280 4.1.1.2 with the `algorithm` object identifiers `id-ml-dsa-44, id-ml-dsa-65, id-ml-dsa-87` specified in RFC XXXX / draft-ietf-lamps-dilithium-certificates:

```
id-ml-dsa-44 OBJECT IDENTIFIER ::= { joint-iso-itu-t(2)
    country(16) us(840) organization(1) gov(101) csor(3)
    nistAlgorithm(4) sigAlgs(3) id-ml-dsa-44(17) }

id-ml-dsa-65 OBJECT IDENTIFIER ::= { joint-iso-itu-t(2)
    country(16) us(840) organization(1) gov(101) csor(3)
    nistAlgorithm(4) sigAlgs(3) id-ml-dsa-65(18) }

id-ml-dsa-87 OBJECT IDENTIFIER ::= { joint-iso-itu-t(2)
    country(16) us(840) organization(1) gov(101) csor(3)
    nistAlgorithm(4) sigAlgs(3) id-ml-dsa-87(19) }
```

and a `parameters` field of type `NULL`, as specified in draft-ietf-lamps-dilithium-certificates .

9.4.4 subjectPublicKeyInfo

The `subjectPublicKeyInfo` field (8.7) conforms to the general ASN.1 structure specified by RFC 5280 4.1 with the `algorithm` object identifiers `id-ml-dsa-44`, `id-ml-dsa-65` and `id-ml-dsa-87`:

```
id-ml-dsa-44 OBJECT IDENTIFIER ::= { sigAlgs 17 }
id-ml-dsa-65 OBJECT IDENTIFIER ::= { sigAlgs 18 }
id-ml-dsa-87 OBJECT IDENTIFIER ::= { sigAlgs 19 }
```

and a `parameters` field of type `NULL`, as specified in RFC XXXX / draft-ietf-lamps-dilithium-certificates.
The `subjectPublicKey BIT STRING` encapsulates the Raw Byte String ML-DSA-44-PublicKey, ML-DSA-65-PublicKey, ML-DSA-87-PublicKey, as specified in draft-ietf-lamps-dilithium-certificates:

```
pk-ml-dsa-44 PUBLIC-KEY ::= {
  IDENTIFIER id-ml-dsa-44
  -- KEY no ASN.1 wrapping --
  CERT-KEY-USAGE
    { digitalSignature, nonRepudiation, keyCertSign, cRLSign }
  PRIVATE-KEY ML-DSA-44-PrivateKey }  -- defined in Section 6

pk-ml-dsa-65 PUBLIC-KEY ::= {
  IDENTIFIER id-ml-dsa-65
  -- KEY no ASN.1 wrapping --
  CERT-KEY-USAGE
```

```
    { digitalSignature, nonRepudiation, keyCertSign, cRLSign }
  PRIVATE-KEY ML-DSA-65-PrivateKey }  -- defined in Section 6

pk-ml-dsa-87 PUBLIC-KEY ::= {
  IDENTIFIER id-ml-dsa-87
  -- KEY no ASN.1 wrapping --
  CERT-KEY-USAGE
    { digitalSignature, nonRepudiation, keyCertSign, cRLSign }
  PRIVATE-KEY ML-DSA-87-PrivateKey }  -- defined in Section 6


ML-DSA-44-PublicKey ::= OCTET STRING (SIZE (1312))

ML-DSA-65-PublicKey ::= OCTET STRING (SIZE (1952))

ML-DSA-87-PublicKey ::= OCTET STRING (SIZE (2592))
```

9.4.5 signatureValue

The `signatureValue` field (8.9) encodes the result of applying the signing algorithm as a `BIT STRING`.