MLS-DSA / ML-KEM Certificates I-Ds

IETF 122 - LAMPS WG

Jake Massimo, Panos Kampanakis, **Sean Turner**, & Bas Westerbaan

Datatracker: draft-ietf-lamps-dilithium-certificates & draft-ietf-lamps-kyber-certificates
GitHub: ML-DSA Certificates & ML-KEM Certificates

ML-KEM Certificates

New version; see diff

Some reorganization

Updated ASN.1 & text from the Great Private Key War of ‘25™; see issue #95

    Text about ASN.1 & consistency checking

Updated Examples

Need to merge: Align with ML-DSA I-D; see PR#104.

Great Private Key War ‘25: ML-KEM

———

```
pk-ml-kem-512 PUBLIC-KEY ::= {
  IDENTIFIER id-alg-ml-kem-512
  -- KEY no ASN.1 wrapping; 800 octets --
  PARAMS ARE absent
  CERT-KEY-USAGE { keyEncipherment }
  PRIVATE-KEY ML-KEM-512-PrivateKey
  }
```

```
ML-KEM-512-PrivateKey ::= CHOICE {
  seed [0] OCTET STRING (SIZE (64)),
  expandedKey OCTET STRING (SIZE (1632)),
  both SEQUENCE {
      seed OCTET STRING (SIZE (64)),
      expandedKey OCTET STRING (SIZE (1632))
      }
  }
```

When receiving a private key that contains both the seed and the expandedKey, the recipient SHOULD perform a seed consistency check to ensure that the sender properly generated the private key.

If the check is done and the seed and the expandedKey are not consistent, the recipient MUST reject the private key as malformed.

ML-DSA Certificates

New version; diff.

Updated ASN.1 and text from great Private Key War of ‘25™; see issue #76

    Text about ASN.1 and consistency checking

Updated Examples

Struct HashML-DSA constraint on TLS and on EE certificates for other protocol; see PR#99

Need to merge: Tweak to Private Key Text; see PR#101.

Great Private Key War ‘25: ML-DSA

```
pk-ml-dsa-44 PUBLIC-KEY ::= {
  IDENTIFIER id-ml-dsa-44
  -- KEY no ASN.1 wrapping; 1312 octets --
  PARAMS ARE absent
  CERT-KEY-USAGE { digitalSignature, nonRepudiation,
                   keyCertSign, cRLSign }
  PRIVATE-KEY ML-DSA-44-PrivateKey
  }
```

```
ML-DSA-44-PrivateKey ::= CHOICE {
  seed [0] OCTET STRING (SIZE (32)),
  expandedKey OCTET STRING (SIZE (2560)),
  both SEQUENCE {
      seed OCTET STRING (SIZE (32)),
      expandedKey OCTET STRING (SIZE (2560))
      }
  }
```

When receiving a private key that contains both the seed and the expandedKey, the recipient SHOULD perform a seed consistency check to ensure that the sender properly generated the private key.

If the check is done and the seed and the expandedKey are not consistent, the recipient MUST reject the private key as malformed.

Open Issues: 1st

Do we need text for public key derivation from private?

See issue #93

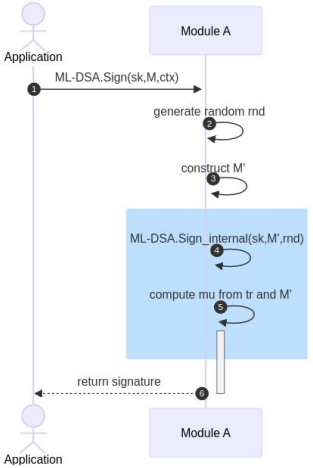
While this would be useful, not entirely sure we absolutely need this in the I-D.

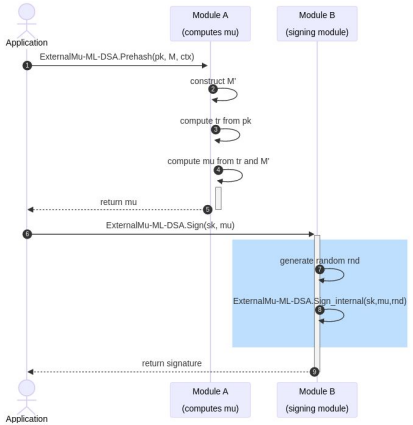
Open Issues: 2nd

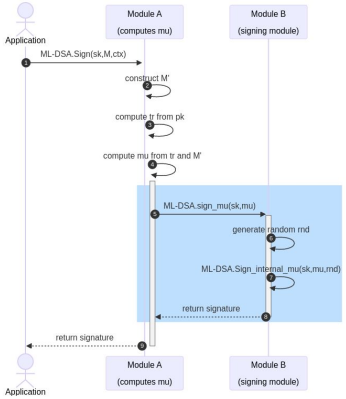
External Mu: Some are concerned about whether what is in the I-D is verifiable by a CAVP/CMVP lab; the mu/ExternalMu-ML-DSA_sign exchanges.

Not sure we need to say anything about this in this I-D:

- Some care about FIPS some don't
- All implementations interop







Picture courtesy of Tim Hudson!

Are we done!?