Use of the ML-DSA Signature Algorithm in the Cryptographic Message Syntax (CMS)

draft-salter-lamps-cms-ml-dsa

Ben Salter, Adam Raine, Daniel Van Geest

IETF 121 – LAMPS

Existing Drafts

	X.509	CMS
ML-KEM	✅	✅
ML-DSA	✅	❌
SLH-DSA	✅	✅
Composite ML-KEM	✅	✅
Composite ML-DSA	✅	✅

Existing Drafts

	X.509	CMS
ML-KEM	✅	✅
ML-DSA	✅	✅
SLH-DSA	✅	✅
Composite ML-KEM	✅	✅
Composite ML-DSA	✅	✅

What's in the draft

- Largely inspired by:
  - *draft-ietf-lamps-dilithium-certificates*
  - *draft-ietf-lamps-cms-sphincs-plus*

- No HashML-DSA – pure mode only

- Specifies ML-DSA-{44,65,87}

- ASN.1 module

Feedback

- Bump specified digest algorithms from RECOMMENDED to REQUIRED
  - Guarantees minimum level of interoperability
  - Aligns with RFC 8419 (EdDSA in CMS)
  - *draft-ietf-lamps-cms-sphincs-plus* should be similarly updated
- Change message digest algorithms from SHAKE to SHA2
  - Same reasoning as for KDF in *cms-kyber* draft – more support for SHA2 at CMS level.
- Maybe do something about EUF-CMA in CMS?
  - See later presentation

Feedback

- Introduce `id-digest-none` for pure {ML,SLH}-DSA signing of non-SignedAttributes?
  - Same discussion in 2017 for draft-ietf-curdle-cms-eddsa-signatures
  - People were fine with the other approach (https://mailarchive.ietf.org/arch/msg/curdle/HG4ED83kc2mf9n4j-O_zL3PGGKg/)
  - Unless people want to reopen the issue, I propose we add explicit text to the *cms-ml-dsa* and *cms-sphincs-plus* drafts:

```
NOTE: Either id-sha512 or id-shake256 is used as part to the
      private key signing operation.  However, the private key signing
      operation does not take a message digest computed with one of
      these algorithms as an input. (RFC 8419)
```