Internet X.509 Public Key Infrastructure: Algorithm Identifiers for Dilithium

draft-ietf-lamps-dilithium-certificates-00
https://github.com/lamps-wg/dilithium-certificates

Jake Massimo, Panos Kampanakis, Sean Turner, Bas Westerbaan

Motivation

- The inclusion of the Dilithium signature algorithm into X.509 certificates.
- This is aimed at the description of “pure” (i.e., non-composite/hybrid) certificates.
- Think of this as RFC 3279 for NIST’s PQ Signature algorithms. The I-D provides the conventions and syntax for putting the algorithm identifiers and parameters into certificates.

Options

- No parameters – the OID tells you all you need to know.
- Key format – OCTET STRING vs BIT STRING.
- One signature algorithm per draft.
- NIST to define OIDs – currently in the draft as TBD.

Updates

Public Keys

- OCTET STRING can be mapped to BIT STRING from RFC 5208.
- “treat BIT STRING and OCTET STRING identically (doing sensible 0-bit-padding if BIT STRING is no multiple of 8)”. https://www.secg.org/sec1-v2.pdf section 2.3.1/2.3.2
- Thus no need for wrapping/adding another TLV layer for implementations that internally operate on octet strings (and tag BIT STRINGS only where the standard mandates it).

Private Keys

- The DilithiumPrivateKey data structure has been modified, a call out to the OneAsymmetricKey structure has been explicitly made.

Updates

Hackathon

- Interop and consistency between numerous LAMPS I-D specifying ASN.1 encodings of Dilithium keys.
- Will continue to meet monthly.

Key Discussion Points

- **Deterministic vs. Randomized** signing
  - Discussion in PQC-forum on new “hedged” mode of Dilithium - randomized signatures that are seeded in part by the message and the key along side a source of randomness.

- **Hash-then-Sign**
  - We are analyzing this in NCCOE and will provide findings. It was also discussed extensively in CFRG WG, LAMPS WG and NIST PQC alias.
  - For this draft, unless something changes to change our minds, we are leaning towards following EdDSA in X.509 RFC8410 which does not prehash for EdDSA.

- **Public Key encoded inside of Private Key**
  - Alternatively, include the values in the private key that allow the reconstruction of the public key for those that want to generate it instead of carrying both keypairs.

- Tracking at https://github.com/lamps-wg/dilithium-certificates/issues

Thank You

Thank you everyone for the discussion and feedback so far! Would love to continue to hear from you for review and feedback.