Algorithms and Identifiers for Post-Quantum Algorithms in the Internet X.509 Public Key Infrastructure

draft-massimo-lamps-pq-sig-certificates-00

Jake Massimo, Panos Kampanakis, Sean Turner, Bas Westerbaan

Motivation

What?

- The inclusion of the algorithms selected by NIST's PQC project into X.509 certificates.
  - A focus on Dilithium as NIST has stated it is the primary algorithm to be implemented and it has well balanced performance.
- 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.

Motivation

Why Now?

- We are seeing industry interest in scoping certificates with PQ resistance. These are applicable in industry sectors that rely on a long-term root of trust that may be embedded within a device or difficult to re-issue.
  - Automotive industry (car ECUs).
  - Heavy machinery.
  - IoT devices (streaming media players) that may sit on a shelf in a warehouse for years before being turned on for the first time.
- NIST has now announced the quantum-secure signature algorithms for standardization (Dilithium, SPHINCS+, FALCON), adoption will take time.
- Aligns with LAMPS charter to “adopt draft for PQC signatures in PKIX certificates”.
- As with the PQ hybrid TLS 1.3 draft, the intention is for this to be ratified after 2024 that NIST will have specified the algorithms.

Goals

- Define data structures for the use of quantum-safe Dilithium signatures algorithms in X.509.
- Clean and concise specification for implementers (e.g. parameters hardcoded in OID).

Non-Goals

- Selecting which post-quantum algorithms to use in X.509.
- Defining OIDs (NIST will do this for us).

Key Discussion Points

- How many algorithms per draft?
  - Multiple algorithms can bloat sections of the standard, particularly if they are built on completely different mathematical systems.
- Which Security levels of each algorithm to include?
  - Everything NIST standardizes? One OID per algorithm, or per algorithm + security level?
- Would love to hear from you for review and feedback.