lETF interim meeting

Post Quantum digital signature use case(s)

Antonio Vaira, Hendrik Brockhaus

Basic Assumptions

- Many products rely on software update processes that need to be secured and kept secure over extremely **long periods of time**.
- Products have very **different life-cycles** and software updates cycles. Therefore not all Post Quantum signature schemes will work with all use cases.
- Signatures of software artifacts can be delivered via **multiple mechanisms**, e.g., using proprietary formats, plain signatures and CMS container.
- Post quantum safe software update mechanisms will be needed to define **migration strategies** towards PQ-algorithms.

We would therefore be interested in covering digital signature use cases, especially for SW updates, exploring limitations and contributing to the definition of protection concepts.

Interesting use case: firmware signing

- **Stateful Hash Based Signature** schemes, which have been already standardized, seem to be a good fit for firmware signing:
  - The limited number of signatures that is possible to perform with a given key pair is not a limiting factor for such use case
  - The size of the private key and the generation time is no issue as long as the state of the private keys are handled professionally, e.g., by a central security infrastructure
  - The size of the signature is among the smallest and it can be verified on embedded devices

- **Stateless** HBS signature schemes and NIST finalists, although not yet standardized, could be relevant for similar use cases.

- The target is to use a common **security infrastructure** to host key pairs used for several use cases, including firmware signing. The benefit of such common architecture is to be able to more easily enforce strong security controls which are crucial with stateful signature schemes.

Our experience so far…

- Today we see several firmware signing mechanisms offered by different chip vendors. The tooling provided by the vendors does not always have standardize input/outputs. Standardization would help with cross-vendor integration, definition of common processes and overall higher security.

- We have started putting together tools and libraries that implement XMSS to make experience with this algorithm and see how it could play out in a security infrastructure. Some aspects of the algorithm (e.g. data format definition) needs to be refined and therefore we have opened an errata.

- As of today we have an RFC describing the usage of LMS in CMS (RFC8708) but nothing for XMSS yet. We do not favor any stateful HBS schema in particular but we are convinced that it can be helpful to continue the standardization effort also for XMSS at least to have a fallback schema. Additionally, standardization activities will favor the support of multiple algorithms for the same security infrastructure.

Open Questions

- Would it make sense for LAMPS WG to start working on a RFC to use also XMSS in CMS or other formats, for example, relevant for firmware signing?
- What were the rationales for the standardization of LMS usage in CMS?
- What is the opinion of the LAMPS WG regarding the NIST finalists, or other HBS schemes like SPHINCS+, and start working on possible standardization?
- A question that goes in a slightly different direction: it appears that stateful HBS schemes can be interesting for long lived X.509 certificate, e.g. root CA certificates. This use case might be interesting for many applications. Should LAMPS WG look into it?