Google's Comments on the NIST SP800-208 Draft Specification

Stefan Kölbl, Roy D'Souza
February 28, 2020

Google anticipates deployment of post-quantum hash-based signature schemes for verified boot, and over-the-air updates, for a range of hardware modules. These modules vary significantly in available power, computational capabilities and related resources.

When deciding between stateless and stateful schemes, for scenarios that are amenable to the larger signature sizes of stateless schemes we would leverage a NIST-recommended scheme, such as the anticipated SPHINCS+. Whereas for other contexts, where it is an imperative to limit signature sizes, we would deploy a NIST-recommended stateful scheme such as LMS/HSS.

Deployment Scenarios

The following three deployment scenarios would most likely be constrained to usage of a stateful scheme:

- Google Security Chips: All Chromebooks are deployed with an embedded Google Security Chip that is candidate for being a quantum-ready hardware root of trust. It would probably have computational abilities similar to an ARM Cortex M3, with limited memory and flash.
- Battery Operated IoT Sensors: These include sensor devices such as Nest Detect, the motion and perimeter sensors used by the Nest Guard secure alarm system. This class of devices has the resource constraints of the previous category, and also needs to operate on the equivalent of an AAA battery for over two years.
- Powered IoT Devices and Chromebooks: These are powered devices based on Intel/AMD and ARM chips, and these lower cost devices have space and other resource constraints that would benefit from compact signatures.

Our choice of stateful hash-based standardization candidates is LMS/HSS, and the following two categories of parameters would be important for addressing the resource constraints of the scenarios outlined above.

Variable (Sub-)Trees

It would be beneficial to have different parameters depending on the level of a multi-tree. The cryptographic modules at a lower level might be deployed in more constrained environments, while a higher-level tree, perhaps belonging to a more trustworthy third party, could afford more expensive computations.

The cadence of firmware updates to devices, even within each category, could differ significantly. A Chromebook might be updated every six weeks, while some IoT devices might only be updated occasionally. Therefore it would be useful to have a choice of parameters for LMS/HSS:

- LMS_SHA256_M24_H5 with LMOTS_SHA256_N24_W8
- LMS_SHA256_M32_H5 with LMOTS_SHA256_N32_W8

- LMS_SHA256_M24_H10 with LMOTS_SHA256_N24_W8
- LMS_SHA256_M32_H10 with LMOTS_SHA256_N32_W8

- LMS_SHA256_M24_H15 with LMOTS_SHA256_N24_W8
- LMS_SHA256_M32_H15 with LMOTS_SHA256_N32_W8

- LMS_SHA256_M24_H20 with LMOTS_SHA256_N24_W8
- LMS_SHA256_M32_H20 with LMOTS_SHA256_N32_W8

- HSS (with 2-4 levels) with any of the above LMS trees at any level.

Security Targets

In the ongoing NIST post-quantum cryptography standardization process five security levels have been defined and the proposed schemes seem to fall into NIST security level 3 and 5, as they do not rely on the collision resistance of the underlying hash function.

In some of our scenarios it might be useful to have variants of LMS/XMSS that target NIST security level 1, as this would provide security comparable to ECDSA with P-256 or Ed25519, while still providing a buffer against quantum adversaries given the limitations of Grover's algorithm (e.g., limited parallelization or that the quantum circuit of the hash functions will be fairly large). Introducing new variants with n = 16 would reduce the signature size for the OTS by over 50%:

- LMOTS_SHA256_N16_W1: 2196 bytes
- LMOTS_SHA256_N16_W2: 1108 bytes
- LMOTS_SHA256_N16_W4: 580 bytes
- LMOTS_SHA256_N16_W8: 308 bytes