ERICSSON

Date: April 14, 2025

Ericsson AB
Group Function Technology
SE-164 80 Stockholm
SWEDEN

Standardization of Additional SLH-DSA Parameter Sets

Dear NIST,

Thanks for your continuous efforts to produce well-written, user-friendly, and open-access security documents. We do not think stateful signatures is a robust solution. We welcome NIST's plan to standardize additional SLH-DSA parameter sets [1—2]. For many use cases, stateless signature parameter sets designed for significantly fewer than 2^{64} signatures pose no issues. We think that NIST should standardize more than one additional parameter set.

Please find below Ericsson's detailed feedback:

- We are only interested in parameter sets that are "hedged", "pure", and use SHAKE256. BSI [3] recommends only the "hedged" variants and prefers the "pure" versions of both SLH-DSA and ML-DSA. Similarly, CNSA 2.0 [4] approves only the "pure" version of ML-DSA. SLH-DSA using SHA2 is considerably more complex than SLH-DSA using SHAKE. Since ML-KEM and ML-DSA are based on SHA-3, it is natural to switch to SHA-3 when migrating to quantum-resistant algorithms instead of sticking to SHA-2 [5-6].

- Regarding security levels, we have strong confidence in standalone SLH-DSA at security level 1 and have higher trust in the security of SLH-DSA-SHAKE-128 than RSA-3072 against classical attackers. We hope that the European Union [7] will align with NIST and the UK NCSC in recommending all security levels of standalone SLH-DSA for industrial use cases. However, since many early government PQC recommendations focus on national security systems, there may be demand for security level 3 or 5.

- In some use cases, private keys used for signing other certificates are used very infrequently. For instance, in our software signing setup, root CA keys are used for approximately 10 signatures over their entire lifetime, while intermediate CA keys are used around 20 times. Fast verification is a key requirement, while key generation and signing speed are not important. Signature size is not a significant concern in our use case. The signing process is manual and requires human intervention.

- End-entity code-signing certificates are used for significantly more signatures than the CA certificates, though still far fewer than 2^{64}. If signing is integrated as an automated step in the build process, producing a few thousand signatures per day, it would take approximately a decade to reach a total of 2^{20}—2^{30} signatures. In such scenarios, where signing performance directly affects build times, the signing time must remain within reasonable limits.

- For software and firmware signing for use in constrained devices or over constrained radio, the signature size can be very important, particularly when dealing with certificate chains. Large signatures can limit the amount of signed data, and the energy cost associated with radio transmission is high compared to computation [8].

- We are not convinced that overuse safety is a critical requirement. In the case of manual signing, any overuse would be deliberate and indicate a compromised CA. For automated signing, we would carefully select parameters to ensure that, even in worst-case scenarios, the number of signatures remains within the defined limits over the certificate's lifetime. Exceeding those limits would be considered a security incident, regardless of any built-in overuse safety. There is also a strong industry trend toward reducing the lifetime of end-entity certificates. In automated signing setups, mechanisms for quickly replacing a compromised key must already be in place. In this context, emphasizing overuse safety may lead to unnecessary performance degradation without offering meaningful security benefits. We believe it would be preferred for NIST to standardize multiple parameter sets (2^{10}, 2^{20}, 2^{30}, 2^{40}) to allow for flexibility based on different use cases.

We support the standardization of the AAA-2/PPP-2 parameter set, as suggested by Jade Philipoom. However, for manual signing of certificates, 2^{10} would be sufficient. We believe 2^{20} is too limited for automatic signing scenarios, such as when signing is integrated into the build process. For such use cases, we think NIST should standardize parameter sets designed to support 2^{30} or 2^{40} signatures, with reasonable signing performance. There is a significant need for flexible parameter sets to support adaptation across diverse industrial use cases. In general, we believe that "additional parameters" would be preferred for all our SLH-DSA use cases.

John Preuß Mattsson,
Expert Cryptographic Algorithms and Security Protocols, Ericsson

References

[1] NIST requests feedback on additional SLH-DSA(Sphincs+) parameter set(s) for standardization
https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/vtMTuE_3u-0/m/LzPijzC9AQAJ

[2] Smaller Sphincs+ or, Honey, I Shrunk the Signatures
https://eprint.iacr.org/archive/2024/018/1737032157.pdf

[3] BSI – Technical Guideline, Cryptographic Mechanisms: Recommendations and Key Lengths
https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines/TG02102/BSI-TR-02102-1.pdf?__blob=publicationFile

[4] The Commercial National Security Algorithm Suite 2.0 and Quantum Computing FAQ
https://media.defense.gov/2022/Sep/07/2003071836/-1/-1/0/CSI_CNSA_2.0_FAQ_.PDF

[5] Ericsson Comments on NIST SP 800-227
https://csrc.nist.gov/files/pubs/sp/800/227/ipd/docs/sp800-227-ipd-public-comments-received.pdf

[6] ML-KEM is Great! What's Missing?
https://csrc.nist.gov/csrc/media/Events/2025/workshop-on-guidance-for-kems/documents/papers/ml-kem-is-great-paper.pdf

[7] EU Coordinated Roadmap for the transition to PQC
https://emanjon.github.io/Slides/2025%20PQC%20side-meeting.pdf

[8] Constrained Radio Networks, Small Ciphertexts, Signatures, and Non-Interactive Key Exchange
https://csrc.nist.gov/csrc/media/Events/2022/fourth-pqc-standardization-conference/documents/papers/constrained-radio-networks-pqc2022.pdf