ERICSSON

Date: May 4, 2026

Ericsson AB
Group Function Technology
SE-164 80 Stockholm
SWEDEN

Comments on SP 800-230 IPD (Additional SLH-DSA Parameter Sets)

Dear NIST,

Thanks for your continuous efforts to produce well-written, user-friendly, and open-access security documents. We welcome NIST's plan to standardize additional SLH-DSA parameter sets. A limit of 2^{24} signatures poses no issues for many use cases. Please find below our comments on the Initial Public Draft of NIST SP 800-230, along with more general comments on NIST's quantum-resistant signatures.

Specific comments on the draft specification:

- While we would welcome even more parameter sets [1], we consider it most important that NIST publishes some additional SLH-DSA parameter sets as soon as possible. Any delay beyond August 2024, when FIPS 203–205 were published, significantly limits the range of use cases in which new algorithms can be deployed during the initial PQC migration. PKI and long-lived devices are high priority areas for migration [2], and SP 800-230 is too late for some already migrated systems [3]. The proposed parameter sets provide substantial value across many use cases. We therefore encourage NIST to publish SP 800-230 without further delay, while remaining open to additional parameter sets and refinements, provided they do not postpone publication.

- *"For many use cases of transmitting and verifying the digital signatures of firmware, software, and digital certificates, those signature sizes and/or their associated verification times are highly undesirable."*

  This statement is accurate; however, "highly undesirable" is context-dependent and does not apply universally. For many infrastructure use cases involving TLS and IPsec, the performance of SLH-DSA as specified in FIPS 205 is adequate. We suggest the more precise formulation: "may be highly undesirable in bandwidth- or latency-constrained environments."

- The draft states that the parameter sets in FIPS 205 are designed to maintain full security for up to 2^{64} signatures. A corresponding statement is missing for the additional parameter sets. We suggest that SP 800-230 explicitly state that the additional SLH-DSA parameter sets maintain full security for up to 2^{24} signatures.

- While the changes in signature size are easy to understand, we suggest that the document also includes a more detailed explanation of how signing and verification performance compares to the already standardized "s" and "f" variants in Table 2 of FIPS 205 [4].

- The specification clearly explains that users must perform a thorough evaluation to ensure that the signature limit is never exceeded over the lifetime of any signing key. We suggest that the document also explicitly state that no such limits apply to the verification key, as this clarification would likely be helpful to readers.

- While the primary advantage over LMS and XMSS is that SLH-DSA is stateless, the specification should also discuss its advantages relative to ML-DSA and FN-DSA. In particular, the main advantage of SLH-DSA lies in its security assumptions: it relies only on the strength of the underlying hash function, whereas ML-DSA and FN-DSA additionally rely on the hardness of underlying lattice problems.

- We suggest stating explicitly that SLH-DSA shall be instantiated as specified in Section 11.1 of FIPS 205 [4] for the SLH-DSA-SHAKE-128-24, SLH-DSA-SHAKE-192-24, and SLH-DSA-SHAKE-256-24 parameter sets; as specified in Section 11.2.1 for the SLH-DSA-SHA2-128-24 parameter set; and as specified in Section 11.2.2 for the SLH-DSA-SHA2-192-24 and SLH-DSA-SHA2-256-24 parameter sets.

- We suggest expanding Table 1 to include a column for the signature limit and assigning this parameter a variable name. This would facilitate combining parameter sets from FIPS 205 and SP 800-230 into a single table, where inclusion of the signature limit is necessary, and would help establish a consistent term for this parameter.

- *"these cached values is permissible and does not affect the security properties of the algorithm"*

  The document requires that users perform an evaluation of the operational environment, including the maximum potential speed of each signing operation. As cached values affect the maximum potential speed of each signing operation, it may impact the ability to guarantee the 2^{24} signature limit. We suggest that this is explicitly noted in the specification.

- We suggest the document explicitly note that no attacks on strong unforgeability (SUF-CMA) are known for SLH-DSA, and that the scheme is widely believed to provide SUF-CMA security, i.e., given multiple message-signature pairs (m_i, σ_i) an attacker cannot create a new signature (m_i, σ') for any previously signed message m_i. Signature schemes with trivial attacks on strong unforgeability are unsuitable for general-purpose use and have resulted in serious practical security vulnerabilities in real-world systems. Many developers and systems treat a message-signature pair as "something the signer actively emitted," rather than "the signer once signed this message". This can undermine system integrity, auditability, and provenance, and, in

the worst case, may even enable replay attacks. See the general comments on pages 3–4 for a discussion of the importance of phasing out malleable signature schemes with trivial attacks on strong unforgeability.

– We suggest the document explicitly state that SLH-DSA is designed to be robust against multi-key attacks, i.e., attacks on any one of many public-private key pairs.

General comments on NIST's quantum-resistant signatures:

We consider SP 800-227 Recommendations for Key-Encapsulation Mechanisms [5], to be an excellent document, and we believe NIST should develop a similar publication for signature schemes. Modern signature algorithms introduce several new and important characteristics that users should be aware of, and these differences warrant additional guidance from NIST:

– FIPS 204 and FIPS 205 introduce several concepts that are likely new to many users, such as beyond-unforgeability (BUFF) properties [6–7], the context string ctx, the use of the seed ξ as a private key, hedged and deterministic hashing, pre-hashing via external-μ, HashML-DSA and HashSLH-DSA, and protocol-layer pre-hashing where a digest is used as the message M [8]. Future standards such as FN-DSA [9] may explore an expanded feature set, including, for example, key and message recovery, identity-based and blind signatures, threshold and aggregate signatures, and zero-knowledge-friendly constructions. It is preferable to provide guidance on these types of mechanisms in a shared specification, rather than repeating them across FIPS 204, 205, 206, and potential future NIST specifications for signature algorithms such as MAYO, SQIsign, and FAEST.

– All modern signature schemes (RSA-PSS, EdDSA, LMS, XMSS, ML-DSA, SLH-DSA, FN-DSA) avoid trivial attacks on strong unforgeability and are widely believed to provide a high level of SUF-CMA security [10]. The transition to PQC provides an excellent opportunity to phase out signatures with trivial attacks on strong unforgeability. Malleable EUF-CMA signatures have enabled serious attacks in the past [11] and will likely do so again if they continue to be used. If any future signature schemes are standardized despite known low-complexity attacks against SUF-CMA security, such schemes should be clearly labeled with appropriate warnings and should not be considered general-purpose.

  Malleable EUF-CMA signatures can undermine system integrity, auditability, and provenance, and, in the worst case, may even enable replay attacks, see [12]. This is also true for certificates. There is a significant gap between what people think a certificate fingerprint represents and what cryptography actually guarantees when a certificate is signed with a malleable EUF-CMA only signature algorithm. With such signature algorithms, a CA does not issue a single certificate; instead, it issues a set of valid certificates, each with its own fingerprint. This mismatch has practical consequences. Logging, SIEM, and threat intelligence systems often record events such as "Observed certificate fingerprint X connecting to service Y," implicitly treating the fingerprint as a stable identifier. Similarly, some firewall blocklists operate on fingerprints (e.g., "Block fingerprint X"), see e.g., [13], and incident response workflows often rely on fingerprints as unique identifiers when searching for the attacker across datasets. In the presence of only EUF-CMA guarantees, these assumptions break down, as the same underlying

certificate can appear under many fingerprints. That is, given a certificate (tbsCertificate, Signature), an attacker can derive one or more additional valid certificates (tbsCertificate, Signature′) that have different fingerprints H(tbsCertificate, Signature). Nation-state attackers are known to deliberately confuse logging systems, evade signature-based detection, and introduce ambiguity into forensic analysis, and they can be expected to exploit signature malleability for these purposes as well.

We believe that standardizing ECDSA with trivial attacks on strong unforgeability was a mistake that should not be repeated. Several widely deployed cryptographic libraries enforce canonical (low-s) ECDSA signatures to eliminate signature malleability. This should be best practice; however, it is not compliant with ECDSA as specified in FIPS 186-5.

– SP 800-227 [5] provides requirements and specifications for hybridizing KEMs. It states that a well-constructed composite KEM should preserve the security properties of its components, which aligns with the European definition [2], where a hybrid is defined as "a combination of a post-quantum algorithm and a quantum-vulnerable algorithm for the same mechanism, such that the security is as high as the stronger of the two components." While hybridization of signatures is generally not considered necessary, some European agencies require temporary hybridization of ML-DSA and FN-DSA during the post-quantum migration period [14]. As no government requires hybridization of SLH-DSA, it is expected that many global industries will use standalone SLH-DSA. For many infrastructure use cases involving TLS and IPsec, the performance of SLH-DSA is adequate.

  For the limited cases where signature hybridization is used, additional NIST guidance would be valuable. We suggest that NIST recommend avoiding signature hybridization altogether, and clarify that any composite signature construction must preserve the security properties of its components. Composite constructions that do not preserve the security properties of ML-DSA should be disallowed from 2035. For niche use cases where hybridization is necessary and EUF-CMA security can be shown to suffice, it is appropriate to use non-composite hybrid approaches based on two independent signatures, each anchored in a separate certificate chain. This approach is currently the only hybridization method that appears sufficiently mature for medium-term deployment. It also offers a clear operational advantage: it avoids the combinatorial explosion problem inherent in composite constructions, and the traditional component can be cleanly removed once it no longer considered to provide meaningful security, which according to NIST is from 2035.

  Composite public keys and signatures cannot be used for long-lived roots of trust in Europe, where the ECCG ACM [15] only considers cryptographic mechanisms as agreed if their underlying cryptographic primitives are agreed. Consequently, composite signatures will likely be deprecated alongside their quantum-vulnerable components. Some poorly designed composite signature constructions such as [16] not only significantly reduce the security properties of ML-DSA by inheriting the malleability of ECDSA, but also introduce additional malleability weaknesses. In particular, when hedged or randomized signing is used, an attacker who has observed n valid signatures of a message M can derive up to $O(n^2)$ distinct new valid signatures for the same message. In practice, repeated signing of the same message M with the same private key ξ often occur when a signing request is retried after failure or interruption, as

well as in high-availability systems where the same message is submitted to multiple HSMs. The composites in [16] also significantly weaken the security of ML-DSA by failing to preserve its beyond-unforgeability (BUFF) properties [6–7]. As shown in [17], existential unforgeability alone does not capture the guarantees required by real-world protocols, and the lack of beyond unforgeability (BUFF) properties in traditional signature schemes has enabled concrete attacks on deployed systems; see [6–7]. We note that there exist signature combiners that preserve both SUF-CMA and BUFF properties [18].

- Stateful signature schemes are unfamiliar to many users, and both the requirement to maintain strict signing state and the requirement that key generation be performed in an HSMs that do not allow secret keying material to be exported introduce opportunities for users to shoot themselves in the foot. We believe these issues should be more clearly and prominently explained to users, with a general recommendation to use stateless signatures.

- It is currently unclear which security categories NIST assigns to the LMS and XMSS parameter sets defined in SP 800-208 [19]. We believe NIST should publish an updated version of SP 800-208 that clearly specifies the security strength associated with each parameter set.

- Key exchange and digital signatures serve fundamentally different use cases and therefore have different security and operational requirements. Hash-based signature schemes are conservative in their security assumptions, whereas lattice-based schemes introduce newer assumptions and may warrant larger security margins in practice. In many applications, signatures are required to provide integrity protection over shorter time horizons than those needed for confidentiality. Signatures can therefore often tolerate lower security category choices than key establishment mechanisms. For example, CNSA 2.0 [20] requires security category 5 for ML-DSA, while permitting all standardized parameter sets for hash-based signature schemes.

- Due to persistent misconceptions that Grover's algorithm could be practically used to attack AES-128 and SHA-256, many users remain hesitant to deploy security category 1 and 2 signatures in new systems. The current statements that AES-128 can continue to be used is insufficient for long-term planning. We suggest that NIST and other government agencies, such as the UK NCSC, provide clearer long-term guidance on the expected viability of security category 1, expressed in a time-bound form. For example, a conservative statement such as "security category 1 will remain approved at least until 2060" would provide useful planning guidance for industry while still allowing flexibility for NIST to extend this horizon if advances in classical computing proceed more slowly than currently anticipated.

John Preuß Mattsson,
Expert Cryptographic Algorithms and Security Protocols
On behalf of the Ericsson Cryptography Team

References

[1] Ericsson comments on Standardization of Additional SLH-DSA Parameter Sets
https://emanjon.github.io/NIST-comments/2025%20SLH-DSA%20parameters.pdf

[2] EU Roadmap for the Transition to Post-Quantum Cryptography
https://ec.europa.eu/newsroom/dae/redirection/document/117507

[3] OpenTitan shipping in production
https://opensource.googleblog.com/2026/03/opentitan-shipping-in-production.html

[4] FIPS 205, Stateless Hash-Based Digital Signature Standard
https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.205.pdf

[5] SP 800-227, Recommendations for Key-Encapsulation Mechanisms
https://csrc.nist.gov/pubs/sp/800/227/final

[6] BUFFing signature schemes beyond unforgeability and the case of post-quantum signatures
https://ieeexplore.ieee.org/document/9519420

[7] A Framework for Advanced Signature Notions
https://eprint.iacr.org/2025/960.pdf

[8] HashML-DSA considered harmful
https://keymaterial.net/2024/11/05/hashml-dsa-considered-harmful/

[9] Falcon Specification version 1.2
https://falcon-sign.info/falcon.pdf

[10] Digital signature schemes with strong existential unforgeability
https://pmc.ncbi.nlm.nih.gov/articles/PMC9925878/

[11] Bitcoin Transaction Malleability and MtGox
https://link.springer.com/chapter/10.1007/978-3-319-11212-1_18

[12] OWASP, Signature Malleability
https://scs.owasp.org/SCWE/SCSVS-CRYPTO/SCWE-054/

[13] Secure Firewall - Blacklisted SSL Certificate Fingerprint
https://research.splunk.com/network/c43f7b49-2dab-4e76-892e-7f971c2f20f1/

[14] ANSSI views on technical aspects of the migration to PQC
https://na.eventscloud.com/file_uploads/b635298de1c10be6d3732863e8b1beca_Day2-1600-ANSSI.pdf

[15] ECCG Approved Cryptographic Algorithms 2.0
https://certification.enisa.europa.eu/document/download/a845662b-aee0-484e-9191-890c4cfa7aaa_en

[16] Composite ML-DSA for use in X.509 Public Key Infrastructure
https://datatracker.ietf.org/doc/html/draft-ietf-lamps-pq-composite-sigs

[17] Seems Legit: Automated Analysis of Subtle Attacks on Protocols that Use Signatures
https://eprint.iacr.org/2019/779.pdf

[18] Bird of Prey: Practical Signature Combiners Preserving Strong Unforgeability
https://eprint.iacr.org/2025/1844.pdf

[19] SP 800-208, Recommendation for Stateful Hash-Based Signature Schemes
https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-208.pdf

[20] The Commercial National Security Algorithm Suite 2.0 (CNSA 2.0)
https://media.defense.gov/2022/Sep/07/2003071836/-1/-1/0/CSI_CNSA_2.0_FAQ_.PDF