ERICSSON

Date: January 23, 2025

Ericsson AB
Group Function Technology
SE-164 80 Stockholm
SWEDEN

Comments on NIST SP 800-56A, 800-56B, and 800-56C Recommendation for Key-establishment Schemes

Dear NIST,

Thanks for your continuous efforts to produce well-written, user-friendly, and open-access security documents. Thanks for your continuous efforts to produce open-access security documents. Please find below our comments on NIST SP 800-56Ar3, 800-56Br2, and 800-56Cr2:

General comments applicable to multiple specifications:

- [ISO/IEC 8825] is a paywalled standard that provides no additional value, as DER encoding is fully specified in the open-access ITU-T X.690 [1]. We believe that security standards hidden behind paywalls are unacceptable [2]. Paywalls significantly discourage security researchers from analyzing and reasoning about standards and their interactions. Open access is critical for security specifications, as history has repeatedly demonstrated that a lack of open discussion and thorough analysis often results in significant vulnerabilities. Paywalled standards also enable selective distribution of weakened standards to specific users, similar to the Swiss Crypto AG affair [3]. Paywalled standards pose a significant cybersecurity risk.

  We strongly recommend that NIST remove all references to paywalled standards and replace ISO/IEC 8825 with the freely accessible ITU-T X.690 [1]. This approach aligns with essential Internet Engineering Task Force (IETF) standards, including RFC 2986, 4210, 4511, 5280, 5652, 6960, 7906, and others, which consistently reference [1].

- The three specifications provide detailed definitions, including security properties, for hash functions, MACs, and PRFs. However, a similar definition for Key Derivation Functions (KDFs) is missing. We recommend that NIST define a KDF as a PRF that maintains collision resistance and preimage resistance even when the key is known. Both KMAC and HMAC satisfy these requirements and should be recommended as general-purpose KDFs.

- Key encapsulation should be included in the definitions, introduction, and other relevant sections.

- The security strength should be aligned with the five new quantum-resistant security categories.

- Deprecated algorithms, such as SHA-1, should be removed.

Comments on 800-56A:

- We strongly support the requirement that *"An ephemeral private key shall be used in exactly one key-establishment transaction."* The practice in some security protocols of reusing private keys while still labeling them as ephemeral is deeply problematic and misleading. Users and developers expect that ephemeral keys are used only once and that their security is independent of sessions involving hostile adversaries. The reuse of ephemeral keys, combined with implementation bugs such as the lack of public key validation, has resulted in exploitable vulnerabilities. These flaws have allowed attackers to recover the so-called "ephemeral" private key, enabling them to completely compromise sessions between legitimate parties. We strongly recommend that the requirement "An ephemeral private key shall be used in exactly one key-establishment transaction" be explicitly included in all NIST specifications on Key Encapsulation Mechanisms (KEMs). Any protocol that reuses private keys should explicitly acknowledge this practice and state that the keys are (semi-)static.

- We think NIST should spend limited time updating 800-56A, as it includes algorithms that are expected to be phased out soon. While X25519 and X448 are critical for hybridizing ML-KEM and should be added to 800-56A, we see no future need for FFDH and Weierstraß curves. Montgomery curves offer superior security and performance and should be the sole choice for ML-KEM hybridization. Hybrids with X25519 have already become the de facto standard for TLS 1.3, DTLS 1.3, QUIC, and SSH. We think NIST should encourage the use of X25519 and X448, as specified in SP 800-186, in hybrid schemes.

  As a less favorable alternative to adding X25519 and X448 to 800-56A, NIST could promote the use of X25519 and X448 as non-approved auxiliary functions in combination with ML-KEM in hybrid schemes. This minor adjustment would still support the adoption of PQC and align with de facto Internet standards.

- We don't think [ANS X9.42] and [ANS X9.63] are to be considered normative references, and we recommend moving them to the informative references section. If any part of [ANS X9.42] or [ANS X9.63] is deemed normative, NIST should revise 800-56A to make sure that [ANS X9.42] and [ANS X9.63] are no longer normative. Paywalled security standards constitute a cybersecurity risk, and we believe NIST should remove all normative references to such standards.

- *"Advanced Encryption Standard (AES) as defined in Federal Information Processing Standard (FIPS) 197,1 and the Keyed-Hash Message Authentication Code (HMAC) as defined in FIPS 1982 make attractive choices for the provision of these services."*

Given the narrow block length of AES, the fact that SHA-2 does not behave like a random oracle, and SHA-2's vulnerability to length-extension attacks, we believe that FIPS 202, SP 800-185, and SP 800-197 are more attractive choices.

Comments on 800-56B:

- We do not think NIST should spend time on updating or revising 800-56B. Both Ericsson and 3GPP agreed with NSA Suite B that integer factorization cryptography should be phased out. While X25519, X448, Ed25519, and Ed448 will be crucial for hybridizing ML-KEM and ML-DSA, we see no future need for integer factorization-based cryptography.

- [ANS X9.44] is not a normative reference and should be moved to informative references.

Comments on 800-56C:

- We welcome NIST's plans [4] to allow hybrid shared secrets of the form $Z' = T \,||\, Z$. However, since many systems are expected to use hybrid shared secrets consisting of more than two components, we recommend that NIST extend support to hybrid shared secrets of the form $Z' = T_1 \,||\, Z \,||\, T_2$, where T_1 and T_2 auxiliary shared secrets. This structure, $Z' = T_1 \,||\, Z \,||\, T_2$ should be secure for all KDFs with acceptable security properties. We anticipate that hybrid shared secrets consisting of more than two components will be relatively common. For instance, [5], authored by the chief cryptographer of the Swedish NCSA, recommends hybrid keying that integrates symmetric keying, post-quantum secure asymmetric keying, and classically secure asymmetric keying. Both TLS 1.3 and IKEv2 support this type of hybrid keying. One concrete example is TLS 1.3 resumption with X25519MLKEM768. Additionally, hybrid keying with X25519, ML-KEM, and BIKE/HQC has been proposed as an alternative to FrodoKEM with better performance.

  We recommend that NIST impose reasonable requirements, such as length limits on T_1 and T_2. There is no justification for allowing megabyte-sized T. It is good that 800-56C already mandates that "the method used to generate T must be known and agreed upon by all parties." The suggestion in [6] to allow hybrid shared secrets of the form $S_1 \| S_2 \| \ldots \| S_t$, where at least one of the shared secrets S_i is NIST approved seems perfect.

- *"When an approved MAC algorithm (HMAC or KMAC)"*

  We suggest revising this to "When an approved hash-based PRF (HMAC or KMAC)," as other approved MAC algorithms, such as CMAC and GMAC, do not meet the required properties [7].

- We recommend that NIST restrict the use of SHA-2 and AES-CMAC as KDFs. Both are unsuitable in scenarios where one of the parties can control the input. SHA-256 and SHA-512 are vulnerable to length-extension attacks, while the issues with AES-CMAC are detailed in [7]. Although SP 800-108 [8] proposes a potential solution for AES-CMAC, it appears ad hoc, lacking both motivation and proof. We believe SHA-2 should be explicitly disallowed as a KDF and removed from the specification. Systems relying on SHA-2 can and should use HMAC. Additionally, we emphasize that NIST should discuss length-extension attacks in FIPS 180-5, as suggested in [9]. For AES-CMAC, we argue that it should not be permitted as a KDF unless it can be demonstrated

that a specific construction and input formatting ensure collision resistance and preimage resistance, even when the key is known. A potential solution could be moving AES-CMAC to an appendix and limiting its use to legacy systems. We note that truncated CMAC is an excellent MAC, which behaves like an ideal MAC, even against multiple forgery attacks.

- We think the specification should talk less about MAC algorithms and instead talk about PRFs. While all PRFs can serve as MACs, a MAC generally cannot serve as a PRF.

- In Section 7.1, "Selecting Hash Functions and MAC Algorithms," we think the recommendation should be to use KMAC whenever possible. AES and SHA-2 have significant problems with side-channels, AES-CMAC and SHA-2 are unsuitable as a KDF when one party can control the input [7], and KMAC offers stronger security properties compared to HMAC. We applaud NIST for mandating the use of SHA-3 in ML-KEM and ML-DSA.

- While Section 8.1, "Using a Truncated Hash Function," correctly notes that truncation may be less efficient than using the corresponding untruncated version, truncation significantly enhances the security properties of algorithms such as SHA-512, CMAC, and HMAC. For example, untruncated SHA-512 provides no protection against length-extension attacks, whereas SHA-384 offers 128-bit security against such attacks. Similarly, untruncated CMAC and HMAC deviate significantly from the behavior of ideal MACs, whereas their truncated counterparts do behave as ideal MACs when the number of queries is limited. A general recommendation should be to use SHA-2, HMAC, and CMAC with a longer-than-required output size and truncate the result. We believe Section 8.1 should either be removed or rewritten to highlight the positive security properties of truncation. Ericsson has historically aligned with NSA Suite B and CNSA 1.0, using (HMAC-)SHA-384 wherever possible. During the PQC migration, we plan to adopt the superior SHA-3 family as widely as possible.

John Preuß Mattsson,
Expert Cryptographic Algorithms and Security Protocols

[1] ITU-T X.690, ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)
https://www.itu.int/rec/T-REC-X.690/

[2] Ericsson Comments on SP 800-38E
https://csrc.nist.gov/csrc/media/Projects/crypto-publication-review-project/documents/decision-proposal-comments/sp800-38e-decision-proposal-comments-2023.pdf

[3] Crypto AG
https://en.wikipedia.org/wiki/Crypto_AG

[4] Ordering of Shared Secrets in SP 800-56C Combiner
https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/ST_yMzYyMl0/m/hEMfXBFHCgAJ

[5] On factoring integers, and computing discrete logarithms and orders, quantumly
http://kth.diva-portal.org/smash/get/diva2:1902626/FULLTEXT01.pdf

[6] NIST SP 800-227 ipd, Recommendations for Key-Encapsulation Mechanisms
https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-227.ipd.pdf

[7] Ericsson Comments on SP 800-38B
https://csrc.nist.gov/csrc/media/Projects/crypto-publication-review-project/documents/initial-comments/sp800-38b-initial-public-comments-2024.pdf

[8] NIST SP 800-108, Recommendation for Key Derivation Using Pseudorandom Functions
https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-108r1-upd1.pdf

[9] Ericsson Comments on FIPS 180-4
https://csrc.nist.gov/csrc/media/Projects/crypto-publication-review-project/documents/initial-comments/fips180-4-initial-public-comments-2022.pdf