ERICSSON

Date: June 1, 2026

Ericsson AB
Group Function Technology
SE-164 80 Stockholm
SWEDEN

Comments on SP 800-52 Rev. 2 (Guidelines for TLS)

Dear NIST,

Thanks for your continuous efforts to produce well-written, user-friendly, and open-access security documents. We welcome NIST's plans to revise SP 800-52. Please find below our feedback on NIST's identified areas of particular concern [1], along with additional comments on SP 800-52 Rev. 2 [2].

- We agree with NIST's proposal to make support of TLS 1.2 optional. As of June 2026, Cloudflare reports that TLS 1.2 accounts for only 2.4% of connections, down from 4.4% a year earlier [3]. The only reason to support TLS 1.2 is interoperability with nodes that are not compliant with the NIST requirement to support TLS 1.3 [2]. Two nodes supporting TLS 1.3 will never negotiate TLS 1.2. While TLS 1.2 can be hardened to provide relatively strong security against classical attackers in use cases where identity protection is not required, most implementations that do not support TLS 1.3 have not been updated for a long time and typically rely on weak cryptography and contain implementation vulnerabilities. In addition to making support for TLS 1.2 optional, we suggest requiring that TLS 1.2 support can be disabled. This aligns with the approach adopted by 3GPP [4] for the 3GPP TLS profile [5]. 3GPP is likely to prohibit support for TLS 1.2 in 6G systems.

- We do not believe there are any compelling reasons to use TLS 1.0 or TLS 1.1 to protect sensitive data. 3GPP has required that clients and servers shall not support TLS 1.0 since 2018 and TLS 1.1 since 2020 [5]. The IETF reached the same conclusion in 2021 [6]. We agree with NSA that these obsolete versions of TLS provide a false sense of security [7]. If TLS 1.0 or TLS 1.1 are used, they should not be considered to provide any meaningful protection. Protection of sensitive data would therefore require security measures at other layers, such as IPsec gateways.

- The TLS 1.2 profile is severely outdated and, for example, permits cipher suites using SHA-1, CBC, and static key exchange. These mechanisms have been prohibited to support by the 3GPP TLS profile since 2020 [5]. However, we do not believe NIST should put effort on updating the

TLS 1.2 profile. Any update to TLS libraries should include support for TLS 1.3, effectively rendering TLS 1.2 and TLS 1.2-specific profiles obsolete.

- The specification is heavily focused on TLS 1.0, TLS 1.1, and TLS 1.2, making it difficult to distinguish guidance relevant to TLS 1.3, which is the only version relevant for future deployments. Some information in the document is also not applicable to TLS 1.3 or is incorrect in the context of TLS 1.3. We suggest that the next revision of the document be refocused on TLS 1.3 and that guidance specific to TLS 1.2 be removed. Accordingly, we limit our detailed comments to TLS 1.3.

- We believe SP 800-52 should also cover QUIC and DTLS 1.3. Modern web communication use a combination of TLS, QUIC, and DTLS rather than TLS alone. TLS 1.3 is used for HTTP/1.1, HTTP/2, and WebSocket; QUIC is used for HTTP/3 and WebTransport; and DTLS is used for WebRTC. As of June 2026, Cloudflare reports that HTTP/3 and QUIC accounts for 32% of all HTTP requests [3]. Specifying requirements and recommendations only for TLS 1.3 therefore provides an incomplete picture of the protocols used to secure modern Web communication.

- *"to protect sensitive data during electronic dissemination across the Internet"*
  "that provides security services over the Internet"

  We suggest reformulating these and similar sentences to avoid references to "the Internet." TLS is used in a wide range of non-Internet use cases, including data centers, enterprise networks, internal communication between processes and components, and 5G core networks, where HTTP/2 over TLS 1.3 is used [8]. The terms "network communications" or "data in transit" are more accurate replacements.

- SP 800-52 Rev. 2 provides limited guidance on modern Mutual TLS (mTLS) configurations, which are a foundational component of Zero Trust Architecture (ZTA), a major priority for U.S. agencies. Explicit mTLS guidance in Rev. 3 would align SP 800-52 with NIST SP 800-207 (Zero Trust Architecture), which identifies mTLS as a key mechanism for implementing zero trust principles across enterprise environments. We suggest that Rev. 3 include specific guidance for mTLS in ZTA environments.

- We suggest that Rev. 3 explicitly state that TLS_SHA256_SHA256, TLS_SHA384_SHA384, and psk_ke shall not be supported. These mechanisms are weak and violate the design principles of TLS 1.3. These mechanisms are prohibited by the 3GPP TLS 1.3 profile [5]. In addition, they are not implemented in BoringSSL or Mozilla NSS, and OpenSSL only exposes them at security level 0, where all insecure options are enabled. Pentagon has stated that symmetric-key establishment mechanisms such as psk_ke will be phased out [9].

- We note that nsa.gov, whitehouse.gov, cia.gov, fbi.gov, and dhs.gov all support X25519 and TLS_CHACHA20_POLY1305_SHA256, while cia.gov and fbi.gov also support X25519MLKEM768. In contrast, nist.gov continues to use an outdated and comparatively weaker configuration that is not compliant with RFC 8446. X25519 provides better performance and security than P-256, and devices without hardware acceleration for AES benefit from ChaCha20-Poly1305.

- We suggest that Rev. 3 state that X25519MLKEM768, as well as standalone ML-KEM, ML-DSA, and SLH-DSA, shall be supported and preferred over quantum-vulnerable algorithms for both the TLS handshake and client and server certificate chains. X25519MLKEM768 has achieved significant real-world deployment and is now the de facto standard for TLS key exchange. Cloudflare reports that X25519MLKEM768 accounts for approximately 70% of all HTTPS requests [10]. Standalone SLH-DSA is well suited for trust anchors, and its performance is also adequate for end-entity certificates in a wide range of infrastructure use cases. For high-availability and real-time systems, SLH-DSA's constant signing time provides predictable performance and latency.

- *"The cryptographic module used by the server **shall** be a FIPS 140-validated cryptographic module [50, 51]. All cryptographic algorithms that are included in the configured cipher suites and the random number generator **shall** be within the scope of the validation."*

  We suggest that Rev. 3 recommend against the use of SecP256r1MLKEM768 and SecP384r1MLKEM1024. Compared to P-256 and P-384, X25519 and X448 offer superior security and performance. A primary motivation for these hybrid constructions is to enable vendors to sell FIPS-validated implementations of P-256 and P-384 as "quantum-resistant," even though only the quantum-vulnerable components are FIPS certified. This practice is highly questionable. We believe the federal government should deploy FIPS-validated ML-KEM.

- We suggest that Rev. 3 prohibit support for composites that weaken the security properties of ML-DSA. ML-DSA is an excellent signature scheme providing SUF-CMA security, beyond-unforgeability (BUFF) properties, and hedged signing. The weak composites in [11] inherit the malleability weakness of ECDSA, introduce a new independent malleability weakness, and destroy the BUFF properties provided by standalone ML-DSA. Compared to standalone ML-DSA, we believe the constructions in [11] are more likely to introduce serious vulnerabilities than to mitigate them. See [12] for a more detailed analysis of [11], including prior real-world vulnerabilities caused by signature malleability and the absence of BUFF properties. As with SecP256r1MLKEM768 and SecP384r1MLKEM1024, a primary motivation for [11] is to enable vendors to sell FIPS-validated implementations of RSA and ECDSA as "quantum-resistant." This practice is misleading, since only the quantum-vulnerable components are FIPS-certified. We believe the federal government should deploy FIPS-validated standalone ML-DSA.

- We recommend that SP 800-52 require RFC 9525-compliant [13] service identity validation. Identity verification failures remain among the most common real-world TLS vulnerabilities. NIST should require verification against the subjectAltName extension and prohibit insecure hostname verification behaviors deprecated by RFC 9525. We further recommend stronger guidance on certificate scope and identity boundaries by discouraging overly broad wildcard certificates and requiring clear service boundary definitions. Such guidance is especially relevant for zero trust architectures. Finally, SP 800-52 should introduce the term "service identity," since TLS often authenticates a service rather than a specific machine.

- We recommend that SP 800-52 explicitly mandate that the TLS implementations make certificate chains and OCSP responses available to the application. Implementations sometimes

do not even support the application retrieving the peer's certificate chain, which renders both certificate chain validation and identity authentication impossible.

– *"TLS servers **shall** be configured with certificates issued by a CA that publishes revocation information in Online Certificate Status Protocol (OCSP) [63] responses."*

  This can be limiting as the WebPKI ecosystem moves away from OCSP. For example, nist.gov currently uses a certificate issued by Let's Encrypt that does not support OCSP [14].

– To prevent authentication from being assumed beyond a certificate's expiration, SP 800-52 should specify that connections shall be terminated once the peer's certificate expires.

– Consistent with the CNSA 2.0 TLS profile [15], we suggest that SP 800-52 state that the early_data extension MUST NOT be used. Aligned with RFC 9846 [16], we suggest that SP 800-52 state that key shares MUST NOT be reused across multiple connections. Reuse of ephemeral keys is already prohibited by SP 800-56A and SP 800-227.

– We suggest Section 1.1 explicitly notes that the TLS 1.3 handshake is based on the SIGMA-I design [17] and provides identity protection, mandatory ephemeral key exchange, and transcript hashing. However, TLS 1.3 unfortunately removed the ability to perform in-band reauthentication and rekeying with Post-Compromise Security (PCS), features that are important for many non-Web applications, including critical infrastructure, mission-critical systems, and space applications. Fortunately, the IETF has started discussions on non-Web use cases of TLS 1.3, DTLS 1.3, and QUIC, including mechanisms to reintroduce in-band reauthentication and PCS [18]. Extended key update does not provide in-band reauthentication; it must instead be handled at the application layer, which requires changes to existing protocols. It is positive that SP 800-52 already forbids Raw Public Keys (RPKs), as they are not suitable for identity validation or zero-trust environments, and their use is also incompatible with the requirements of SIGMA-I.

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

References

[1] NIST Requests Public Comments on SP 800-52 Rev. 2
https://csrc.nist.gov/news/2026/tls-comment-on-sp-800-52-rev-2

[2] Guidelines for the Selection, Configuration, and Use of TLS Implementations
https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-52r2.pdf

[3] Cloudflare Radar Protocol Comparisons
https://radar.cloudflare.com/adoption-and-usage#protocol-comparisons

[4] 3GPP Work Item on Post-Quantum Cryptography Migration
https://www.3gpp.org/ftp/tsg_sa/WG3_Security/TSGS3_127_Malta/Docs/S3-261819.zip

[5] 3GPP TLS profile in Section 6.2 of TS 33.210
https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2279

[6] Deprecating TLS 1.0 and TLS 1.1
https://www.rfc-editor.org/rfc/rfc8996.html

[7] Eliminating Obsolete Transport Layer Security (TLS) Protocol Configurations
https://media.defense.gov/2021/Jan/05/2002560140/-1/-1/0/ELIMINATING_OBSOLETE_TLS_UOO197443-20.PDF

[8] TS 33.501 Security architecture and procedures for 5G System
https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3169

[9] Pentagon Memorandum on Preparing for Migration to Post Quantum Cryptography
https://dowcio.war.gov/Portals/0/Documents/Library/PreparingForMigrationPQC.pdf

[10] Cloudflare Radar Post-quantum encryption adoption
https://radar.cloudflare.com/post-quantum#post-quantum-encryption-adoption

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

[12] Ericsson Comments on SP 800-230 IPD (Additional SLH-DSA Parameter Sets)
https://emanjon.github.io/NIST-comments/2026%20SP%20800-230%20IPD.pdf

[13] Service Identity in TLS
https://www.rfc-editor.org/rfc/rfc9525.html

[14] OCSP Service Has Reached End of Life
https://letsencrypt.org/2025/08/06/ocsp-service-has-reached-end-of-life

[15] Commercial National Security Algorithm (CNSA) Suite 2.0 Profile for TLS 1.3
https://datatracker.ietf.org/doc/html/draft-becker-cnsa2-tls-profile

[16] The Transport Layer Security (TLS) Protocol Version 1.3
https://tlswg.org/tls13-spec/rfc9846.txt

[17] SIGMA: the 'SIGn-and-MAc' Approach to Authenticated Diffie-Hellman
https://iacr.org/archive/crypto2003/27290399/27290399.pdf

[18] Key Management with Forward Secrecy and Post-Compromise Security
https://mailarchive.ietf.org/arch/browse/km-fs-pcs/