ERICSSON

Date: February 7, 2025

Ericsson AB
Group Function Technology
SE-164 80 Stockholm
SWEDEN

Comments on SP 800-232 Ascon-Based Cryptography

Dear NIST,

Thanks for your continuous efforts to produce well-written, user-friendly, and open-access security documents. We welcome NIST's plans to standardize Ascon-based cryptography.

Our primary concerns with the initial public draft are the absence of an Ascon-based Key Derivation Function (KDF) and the proposed constraints for truncated tags. We would prefer that NIST publish Ascon-AEAD128 alongside an Ascon-based KDF that can be used to key it. Additionally, we believe SP 800-232 should not be published with the currently suggested constraints for truncated tags. These constraints should be significantly revised, or truncation should be prohibited entirely.

Please find below our comments on the Initial Public Draft of SP 800-232:

General:

- The specification should clarify that all modes of Ascon are quantum-resistant and specify the security category [1] each mode provide. It would be interesting to know whether nonce masking increases the security level of Ascon-AEAD128 as suggested in [2]. Note that nonce masking does for example not increase the complexity of Grover's algorithm applied to AES in counter mode. Rather than the conventional approach of searching for K such that $AES_K(0^{128}) \oplus P = C$, an attacker can search for K such that $\mathrm{AES}_K^{-1}(P_0 \oplus C_0) \oplus \mathrm{AES}_K^{-1}(P_1 \oplus C_1) = 0^{127} \,||\, 1$. This approach remains effective even with nonce masking, see pages 28—29 of [3]. Similarly, many types of nonce hiding does not increase the complexity of nonce-collision attacks [4].

- We recommend that NIST enhance the introduction by providing a clear, user-friendly explanation of AEAD, hash functions, and XOFs, along with their practical applications. As demonstrated in [5], many developers, end-users, and individuals without a background in cryptography often find these fundamental concepts challenging to grasp.

- *"The family is developed to offer a viable alternative when the Advanced Encryption Standard (AES) may not perform optimally."*

  We think the specification should provide a detailed comparison of the specific benefits and drawbacks of Ascon relative to AES, focusing on aspects such as area, gate count, energy consumption, power efficiency, performance metrics, memory usage, code size, latency, security properties, and key agility. This would likely be valuable to a majority of readers. Although Ascon was designed for efficiency in constrained environments, our understanding is that it provides improved theoretical and practical security properties compared to many modes of AES-128 and SHA-256.

- The specification should explain that Ascon is designed with side-channel resistance in mind, highlighting the critical importance of side-channel protection. Side channels have resulted in numerous exploitable implementation vulnerabilities, both publicly known and undisclosed, and should be a requirement for all new cryptography.

- We welcome many of the changes NIST has made from the Ascon submission v1.2 [6], including allowing 16 rounds, updated initial values, little-endianness, truncation, and nonce masking.

- The terms width and capacity are never used in the specification. We suggest that the term width is removed as it does not add any value over the term state size. The specification should explain the actual numbers for state size, rate, and capacity in the beginning of the specification.

- *"The secret key K and the nonce-masking key K' (if available) shall be generated following the recommendations for cryptographic key generation specified in SP 800-133 [14] and using an approved random bit generator that supports at least a 128-bit security strength."*

  That there is no Ascon-based Key Derivation Function (KDF) as suggested in [7] severely limits the use cases for Ascon. As NIST states, the main feature of Ascon is that it can be used for multiple functionalities, which results in a compact implementation. Without an approved Ascon-based KDF, implementations will most likely use Ascon-CXOF128 as a KDF in a proprietary way. While such proprietary KDFs are likely secure, they may create interoperability challenges in the future. It seems unlikely that constrained systems will also implement Keccak, SHA-2, or AES just to derive keys for Ascon-AEAD128. We would prefer that NIST simultaneously publish specifications for Ascon-AEAD128 and an Ascon-based KDF to key the AEAD. The approach of first publishing Keccak-based hash functions and XOFs in FIPS 202 and then releasing a Keccak-based KDF a year later in SP 800-185 only worked because FIPS 202 exclusively contains unkeyed algorithms.

  The Ascon-based KDF should be secure even when the input secret is non-uniformly distributed, allow variable-length output, and offer collision and pre-image resistance even when the input secret is known [8]. We do not think that a separate Ascon-based DRBG is needed. A KDF has stronger security properties than a PRF, which in turn has stronger security properties than a DRBG or MAC. A single well-designed KDF should be enough for all use cases, but additional functions such as a MAC could be specified if there are significant performance benefits.

- The specification should make it clear that the APIs in Algorithms 3, 4, 5, 6, and 7 are just examples, and that Ascon can be implemented with streaming APIs where the length of the input (A, P, C, M, Z) and the output L are not necessarily available before Ascon is called.

- Appendix B describes how the 64-bit IV is constructed from the parameters in Table 12. We note that as a, b, t, and $r/8$ are uniquely determined by v, which is also included in the IV, there is little value in including a, b, t, and $r/8$ in the IV. Changing this could potentially lead to slightly more optimized implementations.

- The CCM specification SP 800-38C states that a protocol or application should protect against replay attacks, and it has been suggested that NIST should strengthen the recommendations [8]. We think replay protection should be a strong requirement unless careful analysis of the whole system shows that replay protection is not needed in some specific part. Users and developers expect replay protection and higher layer protocols are often designed with the expectation that the security protocol provides replay protection. Systems lacking replay protection are often vulnerable to unexpected attacks and challenging to analyze. If an upper layer was designed with the expectation of replay protection in a lower layer, using a security protocol without replay protection in the lower layers can compromise confidentiality, integrity, and availability in the higher layer, i.e., the whole infosec CIA triad. Practical and serious vulnerabilities due to the lack of replay protection have been common in both standardized and proprietary systems. The specification should strongly recommend that replay protection is used with Ascon-AEAD128.

AEAD:

- We welcome NIST's plan to only standardize a single AEAD based on Ascon.

- *"This section provides an option to implement Ascon-AEAD128 using a 256-bit key, mainly to maintain the 128-bit security strength of Ascon-AEAD128 in a multi-key setting"*

  We think it should be explained already in this section that the 256-bit key does not provide a 256-bit security strength in any setting.

- The specification should state that the nonce masking mechanism is not a nonce hiding transform [9] and state that the masked nonce $N \oplus K'$ shall be secret. A reader might otherwise believe that $N \oplus K'$ can be used as a public field in a security protocol.

- *"Nonce shall be distinct for each encryption operation for a given key to ensure that identical plaintexts encrypted multiple times produce different ciphertext."*

  The specification should describe the security of Ascon-AEAD128 with random nonces. Ascon-AEAD128 with 128-bit random nonces and a fixed key provide the same security as Ascon-AEAD128 with random 128-bit keys and a fixed nonce. This is true even if nonce hiding transforms are used [4].

- In addition to integrity strength, the specification should also describe if Ascon provides reforgeability resistance. This is an essential property, especially for short tags.

- We welcome that NIST uses the term "multi-key setting" instead of the outdated term "multi-user".

- *"The plaintext confidentiality of Ascon-AEAD128 is lost when a nonce is repeated with the same secret key."*

  This makes sense since Ascon-AEAD128 operates like a synchronous stream cipher on the first data block.

  "In the u-key setting, Ascon-AEAD128 with a λ-bit tag provides ($\min\{128-\log2(u), \lambda\}$)-bit security strengths of confidentiality and integrity when a (nonce, associated data) pair is never repeated for two encryptions with each of u keys and the number of nonce repetitions per key for encryption is limited to 2^8"

  We don't see how this is true, considering that confidentiality is lost as soon as a nonce is repeated.

- *"In this scenario, the security strengths of Ascon-AEAD128 are summarized in Table 7."*

  Our interpretation of Table 7 is that it demonstrates the confidentiality against active attackers, which aligns with the integrity strength. However, we would also like to see confidentiality against passive attackers. for a more comprehensive analysis. In practical applications, passive and active attacks are often very different. For audio encryption applications, forgeries may not affect confidentiality.

- *"In the single-key setting, the attacker focuses on a specific key that is shared by one or more users. In contrast, in the multi-key setting with u keys, the attacker aims to compromise any of the u keys used by the users."*

  Multi-key does not imply more users than single-key. A single user can use many keys for protection at rest, and two users can use many keys for protection in transit. In fact, many practical use cases involve one key per encryption invocation, and security protocols between two users following best practices rekey often resulting in a large number of keys, and two users typically have many protocol sessions between them over time. Single-key is mostly a theoretical simplification that does not have much to do with practical security. One suggestion:

  "In the single-key setting, the attacker focuses on a specific key that is shared by one or more users. In contrast, in the multi-key setting with u keys, the attacker aims to compromise any of the u keys used by one or more user."

  But we would instead strongly suggest toning down the single-key scenario in general. Single-key seldom occur in practice and the single-key thinking should be phased out. Attackers will go for most bang-for-the-buck and will do whatever gives most benefit for the least amount of cost. The multi-key setting where the attacker tries to attack a single key is also a theoretical simplification. A multi-key setting (potentially with precomputation) where the attacker aims to compromise a large number of keys, is what most closely corresponds to actual practical attacks. The AT model appears to be a more realistic approach compared to other memory models [10]. Precomputation can be seen as the cost of designing the VLSI chip. Ignoring the difference in value between various data, the important real-world property is the average cost per recovered plaintext byte.

- *"When the tag bit length is λ, $64 \leq \lambda \leq 128$, the maximum number of decryption failures for a fixed key shall be at most $2^{\lambda-64}$"*

  We welcome that the specification allows 64-bit tags, which is a strong requirement in many constrained use cases. However, availability is an essential part of any security systems (the A in the CIA triad) and enforcing a small limit of the number of decryption failures create denial-of-service problems and implies very low availability and robustness [11–12]. A low limit on decryption failures makes Ascon with 64-bit tags completely unusable in most applications. An application might derive a new key for each invocation, but then the requirement should be 1 invocation rather than 1 decryption failure.

  If Ascon with 64-bit tags is used for file encryption of several files, a single decryption failure would mean that the application is forbidden to decrypt any of the other files. It is hard to understand why NIST allows a constrained application to encrypt things with AES-CTR or AES-CBC but effectively forbids from using Ascon with 64-bit tags.

  We do not think SP 800-232 should be published with the current constraints for truncated tags.

- *"The key shall be updated to a new one when the total number of input data blocks or the number of decryption failures reach their respective limits or if the nonce uniqueness requirement is violated."*

  Why and how does the implementation keep track of nonce uniqueness requirements? On the encryption side the implementation should avoid nonce reuse instead of detecting nonce reuse and then rekeying. On the decryption side, the implementation should perform replay protection, allowing it to simply ignore nonce reuse.

  Doing rekeying too early before the confidentiality or integrity of the algorithm decreases significantly faster than linear typically increases the multi-key advantage for the session. The exact multi-key advantage depends on the algorithm but can be as much as u times its single-key advantage where u is the number of keys [11]. Does the rekeying based on decryption failures improve practical security in any way?

  Enforcing rekeying based on single-key advantages transform the setting to a multi-key setting, invalidating the single-key advantages [11]. The specification does not list the confidentiality advantages against passive attackers, integrity advantages in the single-key setting, or how a successful forgery affects the probability of subsequent forgeries. It is therefore unclear how NIST motivates the requirements and how they are supposed to improve practical security. Earlier NIST requirements for rekeying based on decryption failures have been based on the incorrect assumption that tag length is a good measure of security and the incorrect assumption that rekeying improves practical security [8].

  We do not think SP 800-232 should be published with the current rekeying requirements based on decryption failures and nonce uniqueness requirement.

Hash Functions and XOFs:

- We welcome that NIST standardizes a customized XOF. We believe that Ascon-CXOF128 is far more useful than Ascon-Hash256 and Ascon-XOF128, but our understanding is that Ascon-Hash256 and Ascon-XOF128 provide small performance advantages.

- *"This draft standard outlines the technical specifications of Ascon-AEAD128, Ascon-Hash256, Ascon-XOF128, and Ascon-CXOF128, and provides their security properties."*

  Our understanding is that Ascon-Hash256, Ascon-XOF128, and Ascon-CXOF128 provide security against length-extension attacks as well as indifferentiability from a random oracle. The specification should describe these important security properties. Length-extension attacks have resulted in numerous exploitable implementation vulnerabilities.

- Table 9 lists the security strengths of Ascon-Hash256, Ascon-XOF128, and Ascon-CXOF128 as a function of the output length L. The security strength is also a function of the variable length message M. A random function whose input length is len(M) bits cannot provide more than len(M) security against preimage attacks. The preimage security is bounded by the Shannon entropy of the message M. If the message length is known or likely to be short, the preimage security is less than suggested in Table 9. The specification should explain this.

John Preuß Mattsson,
Expert Cryptographic Algorithms and Security Protocols

References

[1] IR 8547 Initial Public Draft
https://nvlpubs.nist.gov/nistpubs/ir/2024/NIST.IR.8547.ipd.pdf

[2] Ascon Team Comments on NIST IR 8547
https://csrc.nist.gov/files/pubs/ir/8547/ipd/docs/nist-ir-8547-ipd-comments-received.pdf

[3] The Multi-User Security of Authenticated Encryption: AES-GCM in TLS 1.3
https://eprint.iacr.org/2016/564.pdf

[4] Collision Attacks on Galois/Counter Mode (GCM)
https://eprint.iacr.org/2024/1111.pdf

[5] PQC Forum email discussion
https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/4_gSpmFccg8/m/eHzw9tkABgAJ

[6] Ascon submission v1.2
https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/spec-doc-rnd2/ascon-spec-round2.pdf

[7] Proposals for Standardization of the Ascon Family
https://csrc.nist.gov/csrc/media/Events/2023/lightweight-cryptography-workshop-2023/documents/accepted-papers/03-proposals-for-standardization-of-ascon-family.pdf

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

[9] “Nonces Are Noticed: AEAD Revisited”
https://eprint.iacr.org/2019/624.pdf

[10] Nice Attacks - but What is the Cost? Computational Models for Cryptanalysis
https://hal.science/hal-02306912v2/document

[11] Hidden Stream Ciphers and TMTO Attacks on TLS 1.3, DTLS 1.3, QUIC, and Signal
https://eprint.iacr.org/2023/913.pdf

[12] Robust Channels
https://eprint.iacr.org/2020/718.pdf