ERICSSON

Date: August 6, 2025

Ericsson AB
Group Function Technology
SE-164 80 Stockholm
SWEDEN

Comments on Accordion Development

Dear NIST,

Thank you for your continued efforts to produce well-written, user-friendly, and open-access security documents. Well-designed accordions with derived functions [1] could provide significantly improved properties compared to many of the cipher modes currently approved by NIST. Beyond very strong cryptographic properties, we believe the derived functions should prioritize usability and usable security. Interfaces and guidelines should be designed to minimize the demands on users and implementers, as well as the adverse consequences of human mistakes [2]. Ideally, usage limits should be high enough that they need not be considered in practice. The practical adoption of the derived functions will heavily depend on performance and other properties. We think the accordion project should design for the next 50 years, not for the most limiting existing AES APIs. The standardization effort should analyze and standardize several accordions with a common set of derived functions.

Please find below Ericsson's detailed comments on NIST proposal [3] to develop three general-purpose accordions:

- *"A cryptographic accordion is a tweakable, variable-input-length strong pseudorandom permutation (VIL-SPRP) constructed from an underlying block cipher*"

  We do not think NIST should define accordions as block cipher modes, even if the three initially proposed accordions are all based on block ciphers. Looking ahead, we hope to see accordions, both in the literature and standardized by NIST, CFRG, based on other primitives such as Keccak, Ascon, AEGIS, or SNOW 5G, which reuse the NIST-standardized derived functions for accordions. Furthermore, support for variable-length tweaks is essential for certain derived functions and should be included in the definition. We propose the following formulation:

  "A cryptographic accordion is a variable-input-length strong pseudorandom permutation (VIL-SPRP) with support for variable-length tweaks, for example constructed from an underlying block cipher"

- We think Acc128, Acc256, and BBBAcc, based on HCTR2 [4], provide a strong starting point if one limits oneself to block-cipher modes. However, it's difficult to assess whether all three are necessary, or whether they are sufficient, until their security, complexity, limits, and performance characteristics are better understood.

- The security of Acc128, Acc256, and BBBAcc are all based AES/Rijndael. A necessity for cryptographic agility is to have a cryptographic primitive to switch to. With the deprecation of Triple DES, NIST does not have a standardized alternative to be used for encryption in the event that AES/Rijndael is broken. As stated in [5], we think NIST should also standardize an accordion based of Keccak [6–7] to enable cryptographic agility.

- We appreciate that NIST is proposing concrete solutions by suggesting HCTR2 as a foundation. However, many high-level questions about the overall goals remain unresolved. Are accordions intended to replace AES-GCM, or are they meant to serve as high-security options for use cases where performance is less critical? If the goal is to replace AES-GCM, then very high performance would be required, likely achievable only through constructions based on the AES round function, such as AEGIS [8], or modern stream ciphers using SIMD instructions, like SNOW 5G [9]. Similarly, NIST has recently proposed standardizing functions for deriving key-nonce pairs for GCM [10]. It remains unclear whether AEAD derived functions for accordions are intended to be used with similar key-nonce derivation mechanisms or to operate independently. If the final total data limits are indeed 2^{48} and 2^{64} bits, the use of a separate key-nonce derivation function will likely be necessary. We recommend that NIST provide updated guidance on the overall goals.

- NIST's requirements for cryptographic accordions [1] states that an accordion should support Beyond-Birthday-Bound (BBB) security to the highest extent reasonable. We note that Acc128, and Acc256 do not support BBB at all, which contradicts these requirements. As we previously stated in [5], we believe the term "beyond-birthday-bound security" should not be used as a requirement. Similarly [1] states that an accordion should support Key-Dependent-Input (KDI) security to the highest extent reasonable, while also stating that KDI security is generally unachievable and unlikely to be relevant for the intended use cases. We recommend that NIST provide updated guidance on the requirements for accordions. Academic research into the multi-key security of HCTR2 would be highly valuable.

- We like the XCTR mode used in HCTR2 [4]. Using a little-endian block counter and XORing it with the value S appears to be an optimal construction for implementing counter mode on modern CPUs.

- *"Acc128 would support an underlying block cipher with 128-bit blocks (i.e., AES). Due to the birthday bound, NIST expects to limit the total data processed under a single key to 2^{48} bits."*

  This is a stricter formulation than IR 8552 [1] that has the requirement of **at least** 2^{48} bits. We would expect this limit to be ≈ 2^{59} blocks = 2^{66} bits. As shown in Section 4 of [11], the entropy loss of a pseudorandom permutation (PRP) in counter mode, which provides an upper bound on the amount of information an attacker can theoretically recover, is $\approx \sigma^2 / 2^b / \ln 4$ where σ is the number of encrypted blocks. This result also apply to the XCTR mode used in HCTR2. We agree with ANSSI [12] and EUCC [13] that $\sigma \lesssim 2^{59}$ blocks is a suitable limit for non-BBB block cipher modes. This appears to be a well-thought-out and balanced requirement, effectively limiting the

amount of information an attacker could recover to no more than $\approx 2^{-10.47} \approx 0.0007$ bits of entropy. NIST has given no motivation for the 2^{48} bits = 2^{41} blocks limit. Unless NIST provides some convincing arguments, we think the limits for Acc128, if standardized, should be 2^{59} blocks.

We also think NIST should follow the recommendation from the Accordion workshop and use byte alignment instead of bit alignment. This considerably reduces complexity in implementation and testing.

- *"BBBAcc would support extended usage beyond the birthday bound with AES. NIST expects the analogous limit on the total data processed under a single key to be at least 2^{64} bits."*

  2^{64} bits = 2^{57} blocks seems low for a mode that claims to offer beyond-birthday-bound security. Can a cipher with a 2^{57}-block limit, below the commonly accepted birthday-bound limit of 2^{59} blocks [11–13], even be called beyond-birthday-bound? 2^{57} blocks falls short of NIST's own definition of BBB security in Section 2.2.2 of [1], which states that the advantage should be negligibly small, even when 2^{64} or more blocks of input are processed. NIST has given no motivation for the 2^{48} and 2^{64}-bit limits. We believe NIST should clarify its rationale and share its reasoning behind the proposed usage limits. While many current NIST specifications for encryption and PRFs should have stricter limits, the suggested limits for Acc128 and BBBAcc seem too strict.

  Furthermore, it would be beneficial for NIST to clarify how it intends to modify HCTR2 in the design of BBBAcc. Is NIST planning to use the approach in [14] or something else? Beyond-Birthday-Bound (BBB) designs typically suffer from lower performance, more complex security proofs, and greater implementation challenges compared to birthday-bound designs. BBBAcc likely needs to use a 256-bit hash function $H_{\overline{h}}$ and a 256-bit block cipher E_k internally, which is complex and overlaps with the much simpler Acc256 design. As stated in [15], *"one must balance the risk of flaws in an increasingly complex implementation with the risk of a cryptanalytic breakthrough. Because more security products fail due to implementation or configuration errors than failures in their underlying cryptographic algorithms, spending limited resources to add cryptographic complexity can at times weaken security rather than improve it."*

- We note that 2^{48} and 2^{64} bits are less than the $\approx 2^{71}$ bits = 2^{64} blocks limit currently allowed in the AES-GCM specification SP 800-38D. One stated motivation for accordions is to process more data than is possible with AES-GCM. With much more restrictive limits than AES-GCM, the value of standardizing Acc128 and BBBAcc is questionable.

- Given an approved 256-bit block cipher, Acc256 is the easiest to analyze with its excellent security, low complexity, and good limits. The only downside is incompatibility with some existing AES implementations and non-optimal performance on other AES implementations. A new universal hash function using a polynomial over GF(2^{256}), or a finite field of similar size, would have to be defined, but this should be relatively straightforward. As Acc256 primarily targets processors that support carryless multiplication, a binary field is likely faster and simpler than a prime field [4]. If development of an accordion is higher priority than the standardization of a wide block cipher, we think an accordion based on Keccak, preferable with twelve rounds, is the most straightforward solution [6–7].

Regarding Rijndael-256, we have trust in Rijndael-256 as currently specified [16]. ETSI SAGE and 3GPP have recently standardized the use of Rijndael-256 for authentication and key generation in MILENAGE-256 [17]. We do not consider related-key attacks to be of practical concern. However we agree with comments that the AES/Rijndael key schedule is non-optimal and could be improved [18]. A redesigned key schedule could offer stronger security properties and potentially even improved performance compared to the current key expansion. In contrast to the NSA [16], we believe a revised key schedule should be non-linear to ensure better diffusion.

Figure 1 illustrates an example key schedule that takes a 128-bit key and generates 256 bytes of pseudo random values that are not very strong as a keystream but strong enough to serve as round keys. The implementation can likely be optimized further. Something similar could be used as a modified Rijndael-256 key schedule.

- The discussion on accordions, Rijndael-256, and PQC algorithms clearly demonstrate that hardware and software APIs which do not expose subfunctions, such as the AES round function and KECCAK-$p[b, n_r]$, greatly limit innovation and can significantly reduce the performance (or security) of future standards. We recommend that NIST strongly encourage implementations to support the AES round function and KECCAK-$p[b, n_r]$. These subfunctions should also be testable for conformance under the Cryptographic Algorithm Validation Program (CAVP).

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

```
void KeyExpansion128(uint8_t* expKey, uint8_t K[16],
uint64_t Tweak, uint8_t n, uint8_t t, uint32_t domain)
{
  __m512i x0 = Tweak||n||t||domain||0;
  __m512i x1 = Tweak||n||t||domain||1;
  __m512i x2 = Tweak||n||t||domain||2;
  __m512i x3 = Tweak||n||t||domain||3;
  __m512i key = load(K);

  x0 = _mm512_xor_si512(x0, key);
  x1 = _mm512_xor_si512(x1, key);
  x2 = _mm512_xor_si512(x2, key);
  x3 = _mm512_xor_si512(x3, key);

  for (int i = 1; i < 8; i++)
  {
    x0 = _mm512_aesenc_epi128(x0, key);
    x1 = _mm512_aesenc_epi128(x1, key);
    x2 = _mm512_aesenc_epi128(x2, key);
    x3 = _mm512_aesenc_epi128(x3, key);
  }

  _mm512_storeu_si512(expKey + 64 * 0, x0);
  _mm512_storeu_si512(expKey + 64 * 1, x1);
  _mm512_storeu_si512(expKey + 64 * 2, x2);
  _mm512_storeu_si512(expKey + 64 * 3, x3);
}
```

Figure 1. Example key schedule that takes a 128-bit key and generates 256 bytes of pseudo random values that are not very strong as a keystream but strong enough to serve as round keys. The implementation can likely be optimized further. Something similar could be used as a modified Rijndael-256 key schedule.

References

[1] Requirements for Cryptographic Accordions
https://nvlpubs.nist.gov/nistpubs/ir/2025/NIST.IR.8552.pdf

[2] NIST Cryptographic Standards and Guidelines Development Process
https://nvlpubs.nist.gov/nistpubs/ir/2016/nist.ir.7977.pdf

[3] PRE-DRAFT Call for Comments: NIST Launches Development of Cryptographic Accordions
https://csrc.nist.gov/pubs/sp/800/197/a/iprd

[4] Length-preserving encryption with HCTR2
https://eprint.iacr.org/2021/1441.pdf

[5] Comments on NIST's Requirements for an Accordion Cipher
https://csrc.nist.gov/csrc/media/Events/2024/accordion-cipher-mode-workshop-2024/documents/papers/comments-on-NIST-reqs-accordion-cipher.pdf

[6] SHAKE modes of operation
https://csrc.nist.gov/csrc/media/Presentations/2023/shake-modes-of-operation/images-media/sess-3-daemen-bcm-workshop-2023.pdf

[7] Deck-Based Wide Block Cipher Modes
https://csrc.nist.gov/csrc/media/Presentations/2023/deck-based-wide-block-cipher-modes/images-media/sess-3-mennink-bcm-workshop-2023.pdf

[8] The AEGIS Family of Authenticated Encryption Algorithms
https://datatracker.ietf.org/doc/html/draft-irtf-cfrg-aegis-aead

[9] SNOW-Vi: an extreme performance variant of SNOW-V for lower grade CPUs
https://eprint.iacr.org/2021/236.pdf

[10] Pre-Draft Call for Comments: GCM and GMAC Block Cipher Modes of Operation
https://csrc.nist.gov/pubs/sp/800/38/d/r1/iprd

[11] Collision-Based Attacks on Block Cipher Modes - Exploiting Collisions and Their Absence
https://eprint.iacr.org/2024/1111.pdf

[12] Guide des Mécanismes cryptoraphiques
https://cyber.gouv.fr/sites/default/files/2021/03/anssi-guide-mecanismes_crypto-2.04.pdf

[13] Agreed Cryptographic Mechanisms
https://certification.enisa.europa.eu/document/download/a845662b-aee0-484e-9191-890c4cfa7aaa_en?filename=ECCG%20Agreed%20Cryptographic%20Mechanisms%20version%202.pdf

[14] A BBB Secure Accordion Mode from HCTR
https://csrc.nist.gov/csrc/media/Presentations/2024/a-bbb-secure-accordion-mode-from-hctr/images-media/sess-8-lee-acm-workshop-2024.pdf

[15] 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

[16] The Rijndael Block Cipher
https://csrc.nist.gov/csrc/media/projects/cryptographic-standards-and-guidelines/documents/aes-development/rijndael-ammended.pdf

[17] Specification of the MILENAGE-256 algorithm set
https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=4243

[18] Pre-Draft Public Comments on SP 800-197
https://csrc.nist.gov/files/pubs/sp/800/197/iprd/docs/sp800-197-pre-draft-public-comments.pdf