ERICSSON

Date: October 24, 2023

Ericsson AB
Group Function Technology
SE-164 80 Stockholm
SWEDEN

Comments on FIPS 202 "SHA-3 Standard" and SP 800-185 "SHA-3 Derived Functions"

Dear NIST,

Thanks for your continuous efforts to produce well-written open-access security documents. We think Keccak is a very useful primitive to build cryptographic algorithms. It is already used in e.g., TUAK [1], Ed448 [2], as well as ML-KEM, ML-DSA, and SLH-DSA [3] and it will likely be used in many future algorithms.

Please find below our comments on FIPS 202 and SP 800-185:

- *"A random function whose output length is d bits cannot provide more than d/2 bits of security against collision attacks and d bits of security against preimage and second preimage attacks"*

  We think the document should also discuss that similar considerations apply to the variable-length input 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 4.

- We think Table 4 should be augmented with the strength against length-extension attacks. It is often claimed that security agencies from different countries participate in standardization and development of security products with the explicit goal of sabotaging security to enhance their surveillance capabilities. It is unfortunately very common that people design their own insecure "keyed hash functions" with SHA-2 by just hashing the key and the message as $\mathrm{H}(K \,||\, M)$. New examples pop up almost every year in the IETF and it likely happens very often in software projects. While we trust SHA-2 and think that it is a recommended set of hash functions, we think it is in NIST's interest to be as open as possible about the limitations of SHA-2. This would increase the trust in NIST as a global SDO for cryptography. As stated by NIST in [4], length-extension is considered a serious attack on a hash function.

- We think the specifications should describe that SHA-3 is designed to provide indifferentiability from a random oracle [5]. This is a property that modern hash functions should provide and from which pre-image, collision, and length-extension resistance follow automatically. It is also a property that many people incorrectly believe all hash functions have and a property that is required for many important uses of hash functions. The security properties listed in FIPS 202 are not enough for uses cases such as PRNGs, PRFs, asymmetric encryption padding, key derivation functions, and signature schemes that require a function producing a "random-looking" output [6]. One example is the use of SHAKE256 as a PRF in ML-KEM [7]. To use the words of John Kelsey [8], we think NIST specifications should align a bit more with the crypto community's proof-oriented worldview instead of the traditional attack-oriented worldview.

- Hardware and software APIs that only support SHA3-224, SHA3-256, SHA3-384, SHA3-512, SHAKE128, and SHAKE256 are very limiting for innovation and might significantly decrease performance (or security) of future standards. One example is ML-KEM, where the use KECCAK-p[1600, 12] would significantly increase performance. We suggest that NIST strongly recommends implementations to support KECCAK-$p[b, n_r]$. Implementations of KECCAK-$p[b, n_r]$ should be possible to test for conformance under the auspices of the Cryptographic Algorithm Validation Program (CAVP) [9], the same applies to the AES round function.

- We think NIST should add TurboSHAKE128, TurboSHAKE256, and KangarooTwelve [10–11] to FIPS 202 and SP 800-185. TurboSHAKE and KangarooTwelve are modes of operation of the KECCAK-p[1600, 12] permutation. We agree with [11] that 12 rounds give significantly increased performance without compromising security. We think TurboSHAKE is a very useful building block for MACs, encryption schemes, KEMs, and signature algorithms. A suggested outcome of the NIST workshop on encryption schemes [12] was that NIST should standardize TurboSHAKE and Rijndael with 256-bit blocks as building blocks for future encryption schemes.

- It would be good if the differences between an extendable-output function (XOF) and a variable-length hash function are explained clearly. If we understand NIST's terminology correctly, the output length of a XOF does not affect the bits that it produces, while the output length of a variable-length hash function do affect the bits that it produces.

- We think FIPS 202 should mention that the fixed-length SHA-3 functions offer meaninglessly high pre-image security significantly hurting their performance. The suggestion from NIST that "preimages need only be as hard to find as collisions" is as correct today as it was then [13]. We think the decision [14] to stick to the original requirements [4] was a mistake. The adoption of SHA-3 was hurt by the perception that it is slow, not the lack of trust [15]. As we can see, the fast SHAKE functions are much more used than the slow fixed-length SHA-3 functions. We think FIPS 202 should recommend modes of (Turbo)SHAKE and write that related outputs in XOFs can be avoided by including the length in the context. We agree with Langley that KangarooTwelve (K12) is the future. KangarooTwelve is extremely fast [16–18] with 0.51 cycles/byte on x86 and 0.75 cycles/byte on ARM. The slow fixed-length SHA-3 functions are mainly for legacy use and could be moved to an appendix.

- Mentioning of SHA-1, 160-bit digest lengths, Triple DES, and 112-bit keys should be removed from FIPS 202 as they are or will be deprecated. We think SHA3-224 should be deprecated together with Triple DES [19], which is the intended use case for SHA3-224.

- It would be good if FIPS 202 refers to SP 800-185 in the introduction and in Appendix A. An overview of all the functions with their high-level properties (fixed length, variable length, XOF, hash-function, PRF/MAC, etc.) would be nice.

Best Regards,
John Preuß Mattsson,
Expert Cryptographic Algorithms and Security Protocols

[1] 3GPP TS 35.231, "Specification of the Tuak algorithm set"
https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2402

[2] IETF RFC 8032, "Edwards-Curve Digital Signature Algorithm (EdDSA)"
https://www.rfc-editor.org/rfc/rfc8032.html

[3] NIST, "Comments Requested on Three Draft FIPS for Post-Quantum Cryptography"
https://csrc.nist.gov/news/2023/three-draft-fips-for-post-quantum-cryptography

[4] NIST, "Announcing Request for Candidate Algorithm Nominations for a New Cryptographic Hash Algorithm (SHA–3) Family"
https://www.govinfo.gov/content/pkg/FR-2007-11-02/pdf/FR-2007-11-02.pdf

[5] Maurer, Renner, Holenstein, "Indifferentiability, Impossibility Results on Reductions, and Applications to the Random Oracle Methodology"
https://eprint.iacr.org/2003/161

[6] Green, "Indifferentiability"
https://blog.cryptographyengineering.com/2012/07/17/indifferentiability/

[7] NIST, FIPS 203 (Draft) "Module-Lattice-based Key-Encapsulation Mechanism Standard"
https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.203.ipd.pdf

[8] Kelsey, "Dual EC in X9.82 and SP 800-90"
https://csrc.nist.gov/csrc/media/projects/crypto-standards-development-process/documents/dualec_in_x982_and_sp800-90.pdf

[9] NIST, "Cryptographic Algorithm Validation Program (CAVP)"
https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program

[10] Bertoni, Daemen, Hoffert, Peeters, Van Assche, Van Keer, Viguier, "TurboSHAKE"
https://eprint.iacr.org/2023/342.pdf

[11] Viguier, Wong, Van Assche, Dang, Daemen, "KangarooTwelve and TurboSHAKE"
https://datatracker.ietf.org/doc/draft-irtf-cfrg-kangarootwelve/

[12] NIST, "The Third NIST Workshop on Block Cipher Modes of Operation 2023"
https://csrc.nist.gov/Events/2023/third-workshop-on-block-cipher-modes-of-operation

[13] Kelsey, "SHA3, Where We've Been, Where We're Going"
https://csrc.nist.gov/csrc/media/projects/hash-functions/documents/burr_dimacs2013_presentation.pdf

[14] Wikipedia, "SHA-3"
https://en.wikipedia.org/wiki/SHA-3

[15] Langley, "Maybe Skip SHA-3"
https://www.imperialviolet.org/2017/05/31/skipsha3.html

[16] Team Keccak, "Is SHA-3 slow?"
https://keccak.team/2017/is_sha3_slow.html

[17] Team Keccak, "Software performance figures"
https://keccak.team/sw_performance.html

[18] Team Keccak, "KangarooTwelve: fast hashing based on Keccak-p"
https://keccak.team/kangarootwelve.html

[19] NIST, "NIST to Withdraw Special Publication 800-67 Revision 2"
https://csrc.nist.gov/News/2023/nist-to-withdraw-sp-800-67-rev-2