Public Comments on the Decision Proposal to Update NIST Special Publication (SP) 800-38E

Comment period: February 8, 2023 – March 10, 2023

On February 8, 2023, NIST's Crypto Publication Review Board announced a proposal to **update** NIST Special Publication (SP) 800-38E, *Recommendation for Block Cipher Modes of Operation: the XTS-AES Mode for Confidentiality on Storage Devices*. The public comments that NIST received on the proposal are collected below.

More information about this review is available from NIST's Crypto Publication Review Project site.

LIST OF COMMENTS

1. Comments from John Preuß Mattson (Ericsson), February 22, 2023

Dear NIST,

Thanks for your continuous efforts to produce well-written open-access security documents. Please find attached our comments on the SP 800-38E Decision Proposal.

Best Regards,
John Preuß Mattsson

ERICSSON

Date: February 22, 2023

Ericsson AB
Group Function Technology
SE-164 80 Stockholm
SWEDEN

Comments on SP 800-38E, Recommendation for Block Cipher Modes of Operation: the XTS-AES Mode for Confidentiality on Storage Devices

Dear NIST,

Unfortunately we are unable to provide detailed comments on NIST SP 800-38E as almost all of the technical details are hidden behind a paywall.

Please find below our high-level comments on SP 800-38E:

- We think security standards behind paywalls are unacceptable. Paywalls strongly discourages security researchers from examining and reasoning about standards and interactions between standards. Open access is very important for security specifications as history has showed over and over again that lack of analysis often lead to significant weaknesses. Many people has argued that paywalls were a major reason why the serious Wi-Fi "KRACK" vulnerability [1], which allowed anyone to get onto a secure network, was undiscovered for so long. Open access is also essential for building trust. People have even been arguing that "paywalls drive mass surveillance." [2]. In two excellent articles [3][4], Anthony Rutkowski analyses the practice of paywall standards and how it creates security vulnerabilities incompatible with the cybersecurity objective:

  > *"Referencing a paywall standard by a standards body or government agency effectively makes them a sales promotion agent."*
  >
  > *"The continued use of ICT standards hidden behind sales point paywalls using glacial development processes is a cyber security vulnerability."*
  >
  > *"It is long past due for the practice to stop and stop enabling it as "a business model"— that is so obviously antithetical to the cybersecurity objective."*

- We strongly suggest that NIST removes all normative references to paywall standards and instead provides an open access specification of XTS-AES in the update to NIST SP 800-38E. The updated SP 800-38E should then be published with a period of public comment. Removing references to paywall standards will improve security and increase trust in NIST as a global standardization organization for cryptography. NIST did a very good thing in FIPS 186-5 by completely replacing paywalled references with an open specification of ECDSA. This change was very well received by the security community. The same day FIPS 186-5 was published a change request was made for the TLS 1.3 specification and approved by the IETF security area director within hours.

- As can be seen in [5], people are implementing XTS incorrectly, likely by using the same tweak i for several sectors or by using a non-primitive element α. One reason might be that they implement based on the Wikipedia description instead of the paywalled IEEE standard. We suggest that the conformance section in NIST SP 800-38E explicitly states that the tweaks i shall be different for each sector and that the primitive element $\alpha = 0\ldots010_2$ shall be used. Even when implemented correctly, XTS have "penguins" in the time dimension, but not in the space dimension like ECB and [5].

- The update of NIST SP 800-38E should discuss the severe weaknesses with XTS-AES. Annex D.2 of the paywalled IEEE Std 1619-2007 is a mixture of design choices and security properties of AES-CTR, AES-CBC, and wide block tweakable ciphers, solutions that were not chosen. Only in Appendix D.4 it is mentioned that the narrow block length of XTS-AES significantly decreases the security. We think many implementers will have a hard time evaluating if XTS-AES is an appropriate solution.

- A significant weakness with narrow blocks and fixed tweaks as in XTS-AES that we cannot find mentioned anywhere is that an attacker in the chosen-plaintext attack model can guess a 16-byte plaintext block and determine if the guess is correct by checking if the new ciphertext block is equal to the old ciphertext block (as the old and new ciphertext blocks use the same fixed tweak). By repeating this an attacker with limited knowledge of the plaintexts can get knowledge of large amounts of plaintexts with relatively low complexity. This type of attack is not possible in systems that forces adversaries to be nonce-respecting such as the use of AES in security protocol like TLS or IPsec where the attacker cannot control the nonce used for encryption.

- It is unfortunate that XTS-AES is the de-facto standard for disk encryption. Narrow block tweakable ciphers like XTS-AES provides no authentication, low levels of confidentiality (IND-CPA, narrow blocks, repeating nonces), and low security against data manipulation and tampering (no authentication, narrow blocks, replay attacks on blocks instead of sectors). For length-preserving encryption, the wide block tweakable cipher Adiantum [6] that is included in the Linux kernel since version 5.0 and in Android since version 10 provides much better security than XTS-AES as the wide blocks (typically 4096 bytes instead of 16 bytes) improves both confidentiality and security against data manipulation. Long term, XTS-AES should be phased out together with most of the IND-CPA modes in SP 800-38A, but CTR is still useful as a fast length-preserving mode when used with signatures. We think it is very good that NIST now states that using ECB constitutes a severe security vulnerability [7]. ECB should have been deprecated a long time ago. Wide block tweakable ciphers like Adiantum should be seen as a last-resort solution when length-preservation is needed. For future disk-based encryption we think NIST should drive the use of authenticated encryption with explicit nonces and replay protection. Modern hard disk drives (HDD) typically use 4 KiB sectors, and modern solid-state drives (SSD) typically use even larger page sizes. With modern AEAD algorithms, integrity protection is almost free, and the overhead of storing a nonce and a tag per 4 KiB sector is negligible. Systems should be updated to handle the overhead of storing a nonce and tag per sector to be able to offer acceptable protection.

- We suggest that NIST limits the approved use cases for XTS-AES to disk sectors in legacy systems that require length-preserving encryption and updates the title of SP 800-38E to reflect this. For everything else, authenticated encryption providing confidentiality against active attackers should be used. The title "… for Confidentiality on Storage Devices" could be understood as XTS-AES is a general-purpose mode for encryption at rest. XTS-AES is not an acceptable general-purpose mode for encryption at rest.

- NIST does not only lack approved wide block tweakable cipher such as Adiantum [6] appropriate for length-preserving encryption, NIST also lacks AEAD modes hardened against nonce misuse such as AES-GCM-SIV [8], XChaCha20-Poly1305 [9], and AEGIS-256 [10], AEAD modes suitable for long plaintexts such as AEGIS [10], as well as one-pass AEAD modes suitable for short tags such as the new modes for AES and SNOW 5G that ETSI SAGE has specified for use in 5G. 192-bit or 256-bit nonces as in [9] and [10] are suitable to use with random nonces and would solve many of the problems identified in the initial public comments received on FIPS 197 and SP 800-38A [11]. AEGIS additionally has the benefit of being significantly faster than AES-GCM for typical Internet payload sizes [12]. We agree with the initial public comments received on FIPS 197 and SP 800-38A [11] that NIST should standardize AES/Rijndael with a block length of 256 bits. AES with 256-bits blocks would have a number of benefits: it can be used with larger plaintexts or more queries without reaching the current 2^{64} birthday bound in AES with 128-bit width, and trivially accept larger nonces/IVs. Rijndael with 256-bit wide block may be used to solve problems such as wide block tweakable ciphers, block/key size extenders (like Coron et al. [13]/TLR/Feistel schemes), hash functions (like Merkle-Damgård/Hirose [14] schemes), with a more decent level of security 2^{128} of indifferentiability. Having a standardized block cipher with 256-bit blocks would also be a valuable primitive for building future schemes such as the ongoing work to construct a 256-bit version of the 3GPP authentication and key generation algorithm MILENAGE. FIPS 197 is already written with variable block lengths in mind and Rijndael with 256-bit blocks can already be implemented quite efficiently on x86-64. Given this we think NIST should standardize AES with 256-bits blocks in the update to FIPS 197 and start a project to standardize new encryption modes (based on the AES round function, Keccak-p, or ASCON) with the above-mentioned properties.

- When updating SP 800-38E, SP 800-38D, 800-38A, FIPS 197, and FIPS 180-4 we think NIST should make it very clear with that all the algorithms are quantum-resistant and will remain secure for decades to come as explained in [15] and [16]:

  > *"Presently envisioned quantum computing architectures typically indicate that the cost per quantum gate could be billions or trillions of times the cost per classical gate."*
  >
  > *"Plausible values for MAXDEPTH range from 2^{40} logical gates (the approximate number of gates that presently envisioned quantum computing architectures are expected to serially perform in a year) through 2^{64} logical gates (the approximate number of gates that current classical computing architectures can perform serially in a decade)"*
  >
  > *"Grover's algorithm requires a long-running serial computation, which is difficult to implement in practice. In a realistic attack, one has to run many smaller instances of the algorithm in parallel"*
  >
  > *"Taking these mitigating factors into account, it is quite likely that Grover's algorithm will provide little or no advantage in attacking AES, and AES 128 will remain secure for decades to come."*

  Someone reading just NISTIR 8319 [17] could easily get the understanding that AES-128 would not be secure if a cryptanalytically-relevant quantum computer (CRQC) is ever built.

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

[1] Matthew Green, "Falling through the KRACKs"
https://blog.cryptographyengineering.com/2017/10/16/falling-through-the-kracks/

[2] Rick Falkvinge, "The recent catastrophic Wi-Fi vulnerability was in plain sight for 13 years behind a corporate paywall"
https://www.privateinternetaccess.com/blog/the-recent-catastrophic-wi-fi-vulnerability-was-in-plain-sight-for-13-years-behind-a-corporate-paywall/

[3] Anthony Rutkowski, "Monumental Cybersecurity Blunders"
https://circleid.com/posts/20220513-monumental-cybersecurity-blunders

[4] Anthony Rutkowski, "Global Standards Collaboration: Is It Possible?"
https://circleid.com/posts/20230203-global-standards-collaboration-is-it-possible

[5] StackExchange, "An XTS penguin
https://crypto.stackexchange.com/questions/58871/an-xts-penguin

[6] Crowley, Bigger, "Adiantum: length-preserving encryption for entry-level processors"
https://eprint.iacr.org/2018/720.pdf

[7] NIST, "Announcement of Proposal to Revise Special Publication 800-38A"
https://csrc.nist.gov/news/2022/proposal-to-revise-sp-800-38a

[8] RFC 8452, "AES-GCM-SIV: Nonce Misuse-Resistant Authenticated Encryption"
https://datatracker.ietf.org/doc/html/rfc8452

[9] CFRG, "XChaCha: eXtended-nonce ChaCha and AEAD_XChaCha20_Poly1305"
https://datatracker.ietf.org/doc/html/draft-irtf-cfrg-xchacha

[10] CFRG, "The AEGIS family of authenticated encryption algorithms"
https://datatracker.ietf.org/doc/html/draft-irtf-cfrg-aegis-aead

[11] NIST, "Public Comments on SP 800-38A and SP 800-38A Addendum"
https://csrc.nist.gov/CSRC/media/Projects/crypto-publication-review-project/documents/initial-comments/sp800-38a-initial-public-comments-2021.pdf

[12] Frank Denis, "BoringSSL AEADs comparison"
https://github.com/jedisct1/openssl-family-bench/blob/master/img/boring-aeads.png

[13] Coron et al., "A Domain Extender for the Ideal Cipher"
https://www.iacr.org/archive/tcc2010/59780270/59780270.pdf

[14] Hirose, "How to Construct Double-Block-Length Hash Functions"
https://csrc.nist.rip/groups/ST/hash/documents/HIROSE_article.pdf

[15] NIST, "Call for Additional Digital Signature Schemes for the Post-Quantum Cryptography Standardization Process"
https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/call-for-proposals-dig-sig-sept-2022.pdf

[16] NIST, "Post-Quantum Cryptography FAQs"
https://csrc.nist.gov/Projects/post-quantum-cryptography/faqs

[17] NISTIR 8319, "Review of the Advanced Encryption Standard"
https://nvlpubs.nist.gov/nistpubs/ir/2021/NIST.IR.8319.pdf

2. Comments from Danny Niu, March 6, 2023

Even though implementations of IEEE-Std.1619-2018 already conform to SP-800-38E, I'd still like to recommend NIST to cite this revision Instead of staying with the old revision. Because:

1. Link rot might happen, and *-2007 might get pulled.
2. Development history are important (non-normative) information
   That some developers, as well as the academics depend on.
3. Citing *-2018 (standalone, or as addition) are the most meaningful
   Revision in terms of content.