Proposals for Standardization of the Ascon Family

John Preuß Mattsson, Göran Selander,
Santeri Paavolainen, Ferhat Karakoç,
Marco Tiloca, Robert Moskowitz

Image by Melanie from Pixabay

Proposals for Standardization

- An important goal: only Ascon for *all* symmetric cryptographic functions
  - to reduce the total area or ROM for *all* cryptographic functions.
- The Ascon family consists of AEADs, fixed- and variable-length hash functions, PRFs, and MAC.
  - To be "appropriate for most forms of tiny tech", much more than AEADs and fixed-length hash functions is needed.
- Having a single cryptographic primitive like Ascon for *all* symmetric cryptographic functions is a desire for many IoT developers and manufacturers.

The Need for a Well-Assorted Ascon Toolbox

- Overlapping modes and standardization at different times creates interoperability problems.
  - A well-assorted and future-proof toolbox of different Ascon modes should be included in the initial NIST draft specification, rather than having them piecemeal added later as done for AES and SHA.
  - Avoid standardizing overlapping Ascon modes.

Proposals for Encryption

Shorter AEAD tags are needed. Variable tag lengths or a set of allowed smaller tag lengths (e.g., 32, 64, 80, 96, 128 bits).

- 32-bit tags are standard in most radio link layers including 5G.
- 64-bit tags are very common in transport and application layers of the Internet of Things.
- 32, 64, and 80-bit tags are also common for protection of audio frames.

Length-preserving IND-CPA encryption. A need for a standardized lightweight IND-CPA encryption without message expansion as an option, i.e., Ascon encryption without an authentication tag.

- Header encryption in DTLS 1.3 and QUIC, field encryption in DRIP, and identity protection in SIGMA-based protocols (e.g., message_2 encryption in EDHOC).
- For the cases that message integrity is provided by a signature such as the use of COSE in the group mode of Group OSCORE and SUIT.

Proposals for Encryption

Omit 160-bit keys. We suggest that NIST redefines the quantum security level I to be based on Ascon-128 and do not standardize Ascon-80pq.

- The NIST PQC project specified their quantum security levels based on symmetrical algorithms where security level I is based on AES-128.
- Using NIST's performance assumptions, one billion CRQCs (estimated to cost one billion USD each) would take a million years of uninterrupted calculation to find a single AES-128 key. The time for finding a single Ascon-128 key would not differ from AES-128 in a practically meaningful way.

Support for longer nonces.

- Increasing the nonce length increases the number of instantiations that can be allowed with a single key and random nonces.
- The limited number of instantiations and the resulting high collision probabilities with random nonces are big problems with AES-GCM.
- We think that these 32 bits that Ascon-128 sets to zero should be used to support 160-bit nonces.

Proposals for Hashing

Variable length hash function only. No need to standardize fixed-length hash functions.

- Anybody needing a 256-bit digest can use a variable-length hash function with $l = 256$.
- The fixed-length SHA-3 hash functions have seen little practical use, while variable-length functions such as SHAKE, cSHAKE, and KMAC have seen significant practical use in implementations as well as in published and upcoming standards.
- Both longer and shorter digests than 256 bits will be needed. Ed25519 does for example require a 512-bit hash. For SHA-2, NIST has afterwards introduced two different variable-length hash functions based on the fixed-length hash functions.
- We strongly think NIST should standardize one of the variable-length hash functions Ascon-Xof or Ascon-Xofa.
- We do not think that NIST should standardize any of the fixed-length hash functions Ascon-Hash or Ascon-Hasha.

Proposals for Hashing

Customizable hash function. We think that NIST should only standardize a customizable variable-length hash function based on Ascon.

- 256-bit zero value used in initialization of Ascon-Xof and Ascon-Xofa can be used to support function-name bit strings and customization bit strings similar to the parameters N and S in cSHAKE.
- Having a customizable variable-length hash function is essential to build algorithms such as Kyber.
- Unless there are significant performance differences, we do not think that NIST should standardize both non-customizable and customizable variable-length hash functions as was done with SHAKE and cSHAKE.

Ed25519 and ECDSA with Ascon. We strongly encourage NIST to specify the use of an Ascon hash function for Ed25519 and ECDSA, preferably already in the initial Ascon draft specification instead of waiting for an update of 186-5.

- Ed25519 currently requires SHA-512, while ECDSA with P-256 can, e.g., be used with SHA-256 or SHAKE128.

Proposals for KDF, PRF, and MAC

Specify KDF, PRF, and MAC. NIST should specify a KDF, a PRF, and a MAC based on Ascon together with an AEAD and a variable-length hash in a single document, instead of in different documents and timepoints as it was done with SHAKE and KMAC.

— The standard should define a KDF that can be used with secrets that are not uniformly distributed, such as ECDH shared secrets. If a PRF can be achieved with fewer passes, the standard should also define a PRF that can be used with secret keys that are uniformly distributed.

— The MAC function should be variable-length. We suggest that NIST standardizes Ascon-Prf, Ascon-PrfShort, and Ascon-Mac. We think that a dedicated MAC function is needed as the Ascon-128 function cannot be used with a fixed nonce.

Duplex mode for key derivation. NIST should consider standardizing the Ascon duplex mode of operation also for key derivation with a suitable interface like "$init()$, $digest = update(M\ , l)$".

— Generalization of a running hash interface "$init()$, $update(M)$, $digest = finalize(l)$".

— The duplex interface maps more naturally to how key derivation is done in modern security protocols, without the need to derive intermediate keys whose only use is being input to the key derivation in the next state of the security protocol.

Proposals for API

API support for Ascon round function.

- New algorithms like AEGIS, Rocca-S, and SNOW 5G makes clever use of the AES round function.
- TurboSHAKE128, TurboSHAKE256, and KangarooTwelve use fewer rounds of the Keccak-p permutation than SHAKE128 and SHAKE256.
- APIs not supporting the AES round function and Keccak-p cannot support acceleration of these new algorithms, which is thwarting innovation.
- NIST should avoid the risk of the same thing happening with Ascon. We think that NIST should at least strongly recommend or even better mandate that implementations support the Ascon round function.