Use of SHA-3 in the Internet Key Exchange Protocol Version 2 (IKEv2) and IPsec

draft-salter-ipsecme-sha3

IETF 121 – IPSECME

4th November 2024

Draft aims

- Specify use of SHA-3 and KMAC/SHAKE:
  - PRFs
  - Integrity algorithms
  - Signature hash functions

Why SHA-3?

- Insurance against any potential weakness in SHA-2
- Smaller cryptographic library if using NIST PQ algorithms
- XOFs are a more modern way to do key expansion

PRFs

- `PRF_HMAC_SHA3_{256,384,512}`
  - Does the obvious thing

- `PRF_KMAC_{128,256}`
  - `prf+` becomes a single call to KMAC XOF, instead of iterated SHA-3 calls
    - Slight change to `prf+` implementation required

AUTHs

- `AUTH_HMAC_SHA3_{256,384,512}_{128,192,256}`
  - Does the obvious thing

- `AUTH_KMAC_{128,256}`
  - The “preferred” way of creating a MAC using SHA3

Signature hash functions

- `SHA3_{224,256,384,512}`
- `SHAKE_{128,256}`

- For use with Digital Signature (14) authentication method

Why not AEADs?

- AEADs *are* generally a better choice for providing authentication, but:
  - Some implementations assume separation of encryption and authentication
  - Those implementations typically use HMAC-SHA2
  - They may prefer to use HMAC-SHA3

- Still useful regardless in PRFs/signatures

Why HMAC *and* KMAC?

- HMAC-SHA3 is a bit ugly but makes migration easier
  - LAMPS are standardising HMAC-SHA3 in CMS

- KMAC is the “modern” way of doing things
  - Using KMAC in `prf+` is more efficient than repeated SHA-3 calls

What now?

- Is this a good idea?
- Do we want both HMAC-SHA-3 and KMAC/SHAKE?
- Does anyone else want to implement this?
  - We’ve implemented it, and have test vectors
- Does the WG want to work on this?