CFRG Small Hashes and KMAC

Guidelines for small hashes
And for KMAC usage
November 11, 2021
Robert Moskowitz
Small hashes

• Hopefully a design compromise
  – Acceptable risk within a constrained environment

• Over cleartext or keyed?
  – Small hashes add risk to key attacks

• Modern hashing hardware has changed the game
  – Math for collision probability is no longer sufficient
Small hashes

• A need for understandable guidelines for designers
  – How to measure risk to hash compromise
  – What is to exposure to attack

• Particular attention to keyed hashes
  – MAVlink 2 for UAS Command and Control has a 6 byte keyed hash for message authentication
Small hashes

• CFRG hash guidelines draft/RFC?
KMAC usage as a keyed hash

- Sadly overlook function
  - $\frac{1}{2}$ the processing cost of HMAC
    - 1 Keccak function vs 2 SHA functions
  - Standardized hash length
    - No discussion on how to truncate hash

- Is there a usage question as FIPS 202 kind of distinguishes between a hash and XOF?
KMAC usage as a KDF

- Of interest here is use with ECDH
- NIST SP800-56Cr1 does NOT recommend KMAC as a 2-step KDF until...
  - Revision of SP800-108, when?
  - But when you look at HKDF and KMAC what is the difference?
    - Need analysis beyond Team Keccak
KMAC usage as a KDF

- KMAC as a KDF is at least \( \frac{1}{4} \) the cost of HKDF
  - 1 Keccak function to 2 HMAC or more
- How to use KMAC for multiple shared secret generation
  - Need 2 128 bit keys, can KMAC \((K,X,256,S)\) split in half yield 2 unique keys of 128 bit strength?
    - Breaking one does not break other?
    - Though 2 KMAC cheaper than doing this with HKDF
    - And how to do key hierarchies
KMAC usage as a KDF

- CFRG led with EdDSA, can it lead with KMAC for broader usage?
  - Note that “once” NIST lightweight crypto competition completes, a LWC equivalent to KMAC is available.
    - Especially if Xoodyak is one of the selected LWC

- CFRG producing guidelines
  - Encouraging KMAC usage
KMAC usage as a KDF

• Less likely bad designs elsewhere
  – Again MAVlink 2:
    • sha256_48(secret_key|(message pieces)|timestamp)
      – We “learned” not to do it that way during the HMAC discussions mid-90s! But it is not in any guidelines.
Thank you for your time
Questions/Comments?