ChaCha20 & Poly1305

Yoav Nir

Adam Langley

draft-nir-cfrg-chacha20-poly1305-06

Agenda

- Why ChaCha & Poly1305
- (Very) brief overview
- What we've done
- Changes since previous time
- Questions ?

Why ChaCha20 & Poly1305

- AES is becoming the only workable algorithm.
  - Need to have an alternative.
- AES is fast with hardware support
  - Not so much without it.
  - No hardware support on low-end Intel and ARM.
- AES is hard to implement securely.
- ChaCha20 & Poly1305 are fast in C implementations.
  - A naïve implementation of ChaCha20 is free of timing side-channels
  - For Poly1305 you need a little bit of advice. Not much.

Brief Overview of ChaCha

constant	constant	constant	constant
key	key	key	key
key	key	key	key
blk-count	blk-count	nonce	nonce

- Constant bytes spell "expand 32-byte k“
- Key is 256 bits (32 bytes)
- 64-bit nonce allows lots of messages
- 64-bit block-count allows up to a million peta-byte message (called a zettabyte)
  - Slightly larger than an IPsec packet or a TLS record.

Brief Overview of ChaCha

- The ChaCha20 runs 80 "quarter-round" operations on this matrix.
- At the end, it's thoroughly scrambled.
- The matrix is serialized and XOR'd with the plaintext.
- The numbers in the matrix, both when deserializing key and counter and when serializing into a byte-stream are treated as little-endian.

What we've done

- Changed the partition of nonce vs block-count
- 96-bits for nonce
- 32-bits for block-count.
- Complies with RFC 5116 requirement.
- Allows 2^{32} senders 2^{64} messages each.
- Reduces maximum size of message to 256 GB
- Still plenty for a TLS record, IPsec packet.

What we've done

- Defined an AEAD construct
  - Poly1305 key generated using ChaCha20 with same key and nonce as used for encryption, and block_count = 0
  - Encryption starting with block_count = 1
  - Poly1305 is calculated over a concatenation of:
    - AAD (protcol specific)
    - Padding to a multiple of 16 bytes
    - Ciphertext
    - Padding to a multiple of 16 bytes
    - Length of AAD (as 64-bit count of octets)
    - Length of ciphertext (as 64-bit count of octets)

Changes since London (version -01)

- More implementation advice.
- More test vectors.
- Added advice about choosing a PRF.
- Modifed AEAD construction for (slight) gain in efficiency.

Questions?