Authenticated Encryption with AES-CBC and HMAC-SHA

draft-mcgrew-aead-aes-cbc-hmac-sha2

mcgrew@cisco.com

Kenny.Paterson@rhul.ac.uk

Authenticated Encryption with Associated Data (AEAD, RFC 5116)

- Inputs
  - Plaintext P
  - Associated Data A
  - Nonce N
  - Key K
- Outputs
  - (Authenticated) Ciphertext

aes-cbc-hmac-sha2

MAC_KEY = initial MAC_KEY_LEN bytes of K

ENC_KEY = final ENC_KEY_LEN bytes of K

S = CBC-ENC(ENC_KEY, P || PS)

T = MAC(MAC_KEY, A || S || AL)

C = S || T

CBC-ENC uses PKCS padding rules

aes-cbc-hmac-sha2 with KDF

MAC_KEY = KDF(K, “MAC”)

ENC_KEY = KDF(K, “ENC”)

S = CBC-ENC(ENC_KEY, P || PS)

T = MAC(MAC_KEY, A || S || AL)

C = S || T

CBC-ENC uses PKCS padding rules

Concat KDF uses HMAC-SHA with AlgorithmID = “ENC” and “MAC”

AEAD algorithms

- AEAD_AES_128_CBC_HMAC_SHA_256
- AEAD_AES_192_CBC_HMAC_SHA_384
- AEAD_AES_256_CBC_HMAC_SHA_512
- AEAD_AES_128_CBC_HMAC_SHA1

Simplifications to json-web-encryption

Current

- CMK, CEK, CIK
- AEAD / non-AEAD
- Header, Encrypted Key, Ciphertext, Integrity Value
- KDF, “kdf”
- Loose ends
  - KDF definition
  - MAC input

Proposed

- CMK
- AEAD
- Header, Encrypted Key, Ciphertext

Proposed actions

- Update json-web-encryption
  - Eliminate non-AEAD encryption
  - Use aead-aes-cbc-hmac-sha2
- Update aes-cbc-hmac-sha2
  - Use Concat KDF (smaller key)
  - Option: CBC use ciphertext stealing
    - More compact ciphertext (0-15 bytes)
    - draft-burgin-kerberos-aes-cbc-hmac-sha2-01