Encrypted CCN for Sessions Draft 2

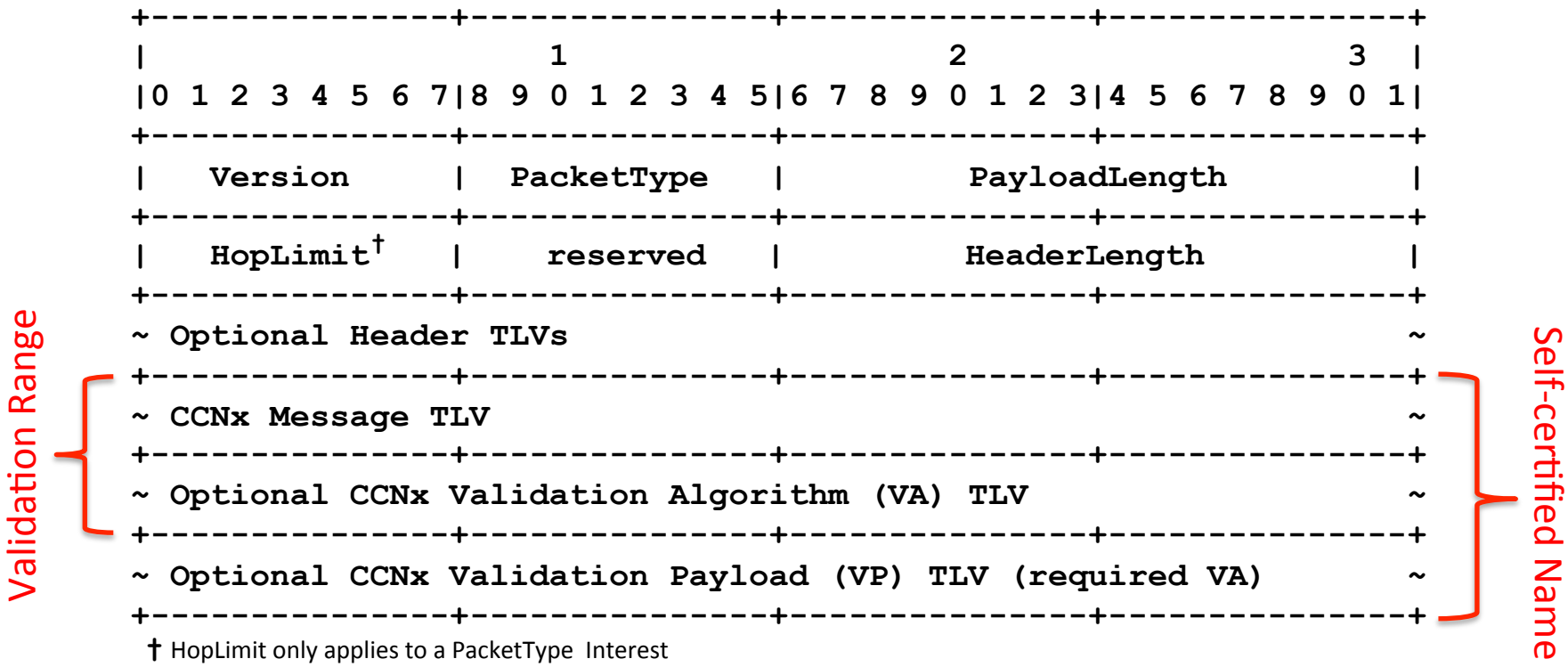
Marc Mosko

June 18, 2015

Problem Area

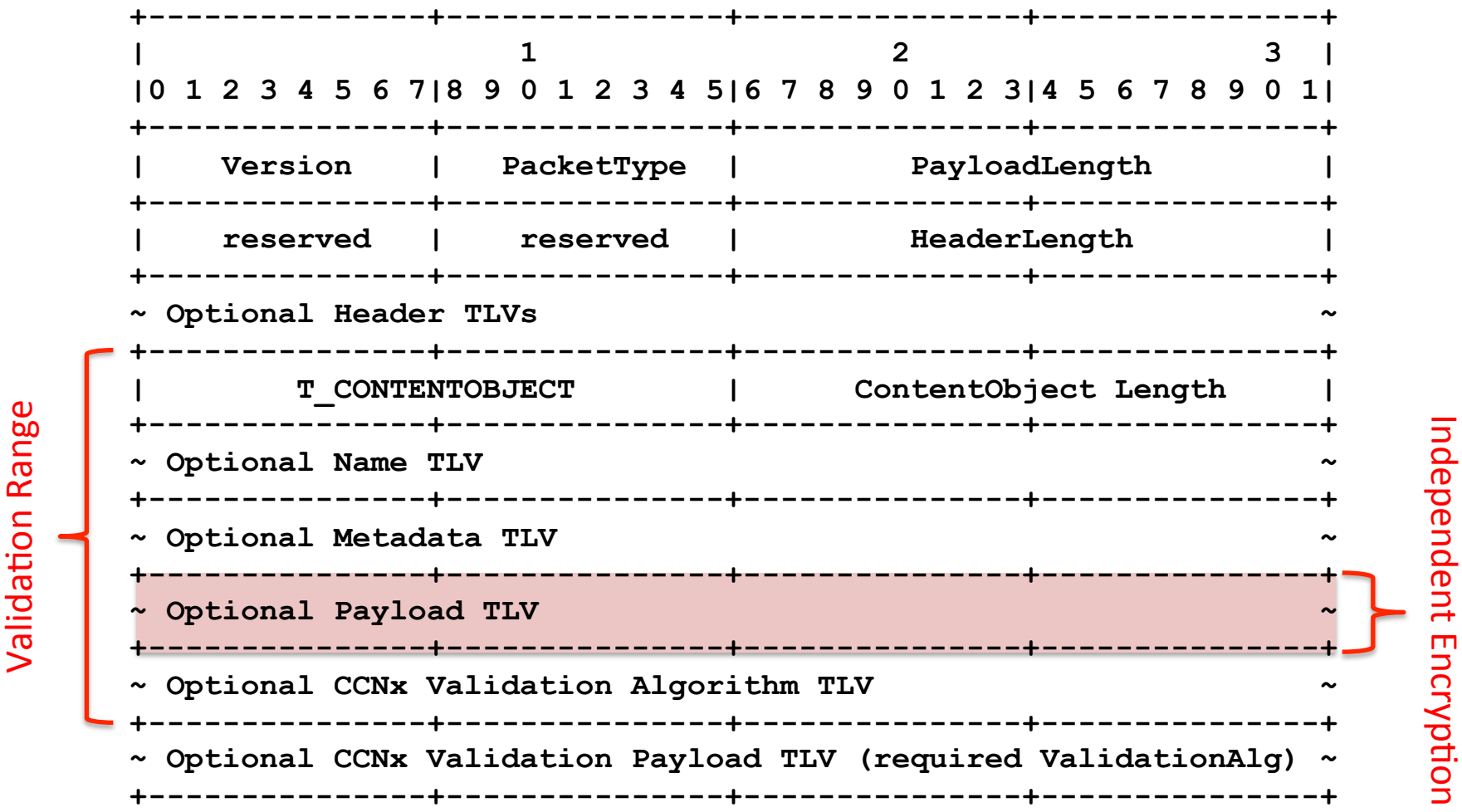
- How to encrypt as much as possible in a CCNx message between systems sharing a secret key.
- Not broadcast encryption.

TLV PACKET



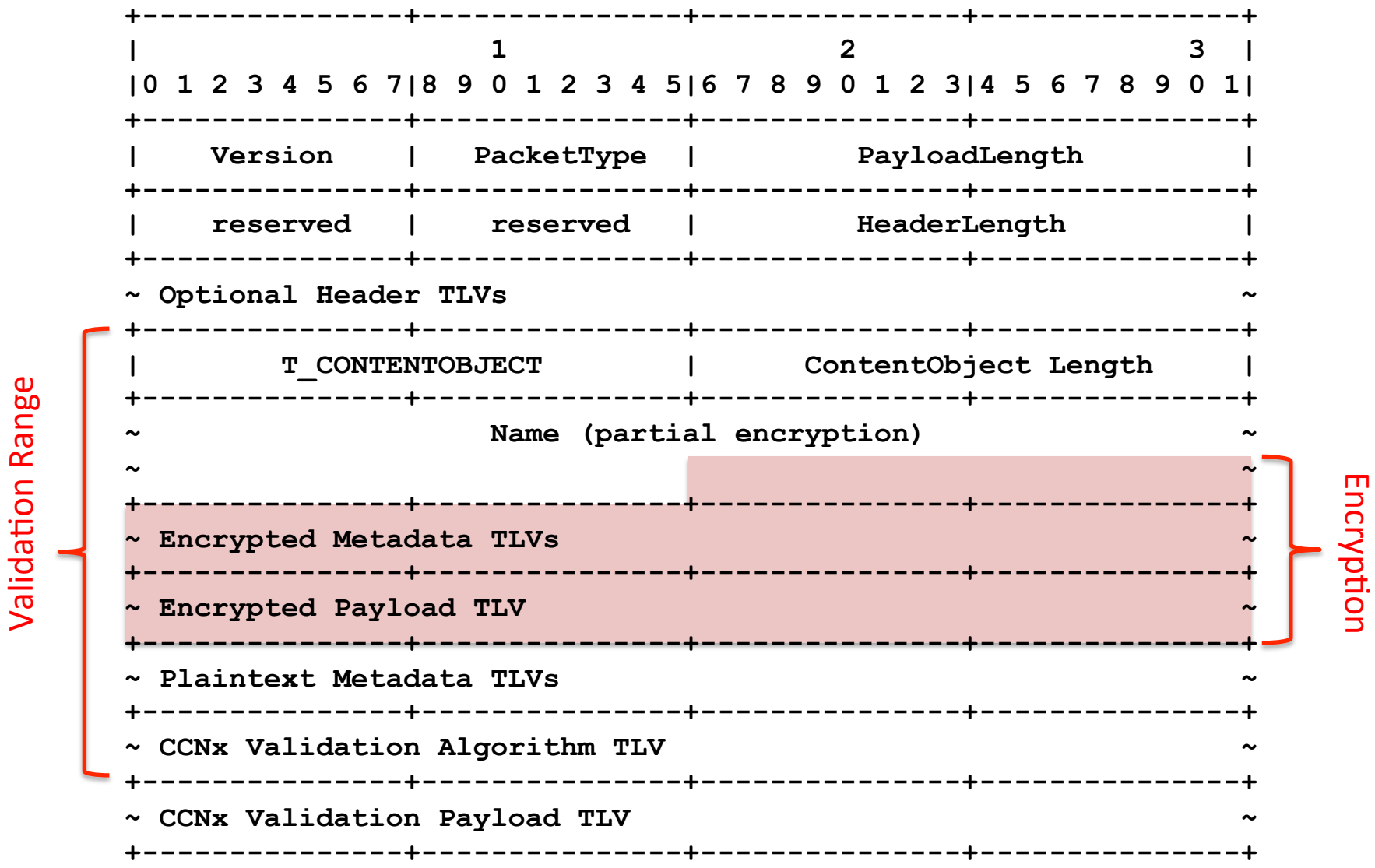
Today's packet format designed around authentication, not necessarily encryption

CONTENT OBJECT



Older ideas around encryption were to only encrypt the Payload and authenticate the whole thing. Independent functions.

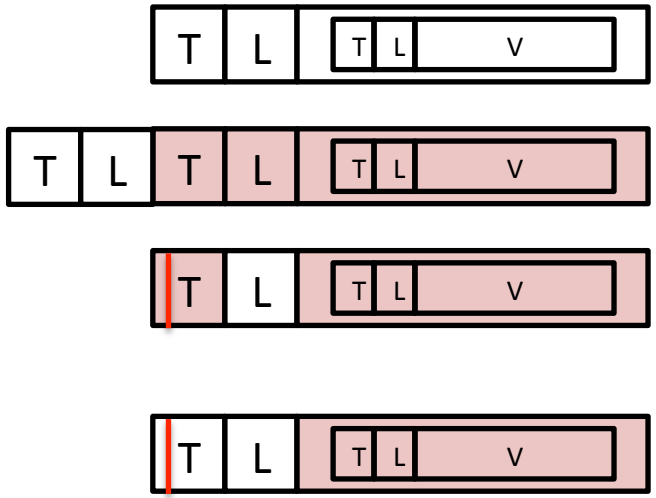
CONTENT OBJECT



We want selective encryption

Encrypting TLV Options

Encryption only applies to first-level TLVs in the CCNx Message.



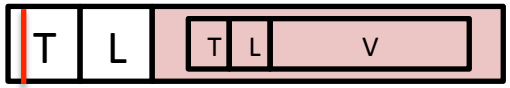
Plaintext

1) Add outer container for encrypted.

2) Encrypt T+V, leave L plaintext, use "encrypted" bit to mark an encrypted T+V.

3) Encrypt V, leave T+L plaintext, use "encrypted" bit to mark an encrypted V. Preferred Option.

Implications of TLV encryption



3) Encrypt V, leave T+L plaintext, use "encrypted" bit to mark an encrypted V. Preferred Option.

- We can begin computing a block cipher from byte 0 of the CCNx message.
- Skip the XOR of all T and L fields.
- Only XOR Vs of Ts marked for encryption.
- Does not require re-ordering fields in CCNx message to have a "plaintext" and "ciphertext" sections.

Authenticated Encryption Associated Data (AEAD)

- AEAD systems allow for an authenticated header plus encrypted plaintext and combine authentication an encryption.
- Used between systems that share the same secret key *K*.
- Still need a key exchange protocol.

Proposed Solution

- CWC AEAD [1]
  - Encrypt-then-authenticate style
  - Given (*A,M*) and nonce *N*, encrypt *M* to *s* with a CTR mode, then use Carter-Wegman MAC with nonce *N* on (*A,s*) to get authenticator *t*. Final message is (*A, s, t*).
  - CWC-AES-kl uses AES-kl as high-speed cypher.
  - Can be parallelized for high speed
  - Uses block cypher with XOR, so no message expansion and 1:1 byte locations, so "L" fields do not change.

[1] http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/proposedmodes/cwc/cwc-spec.pdf

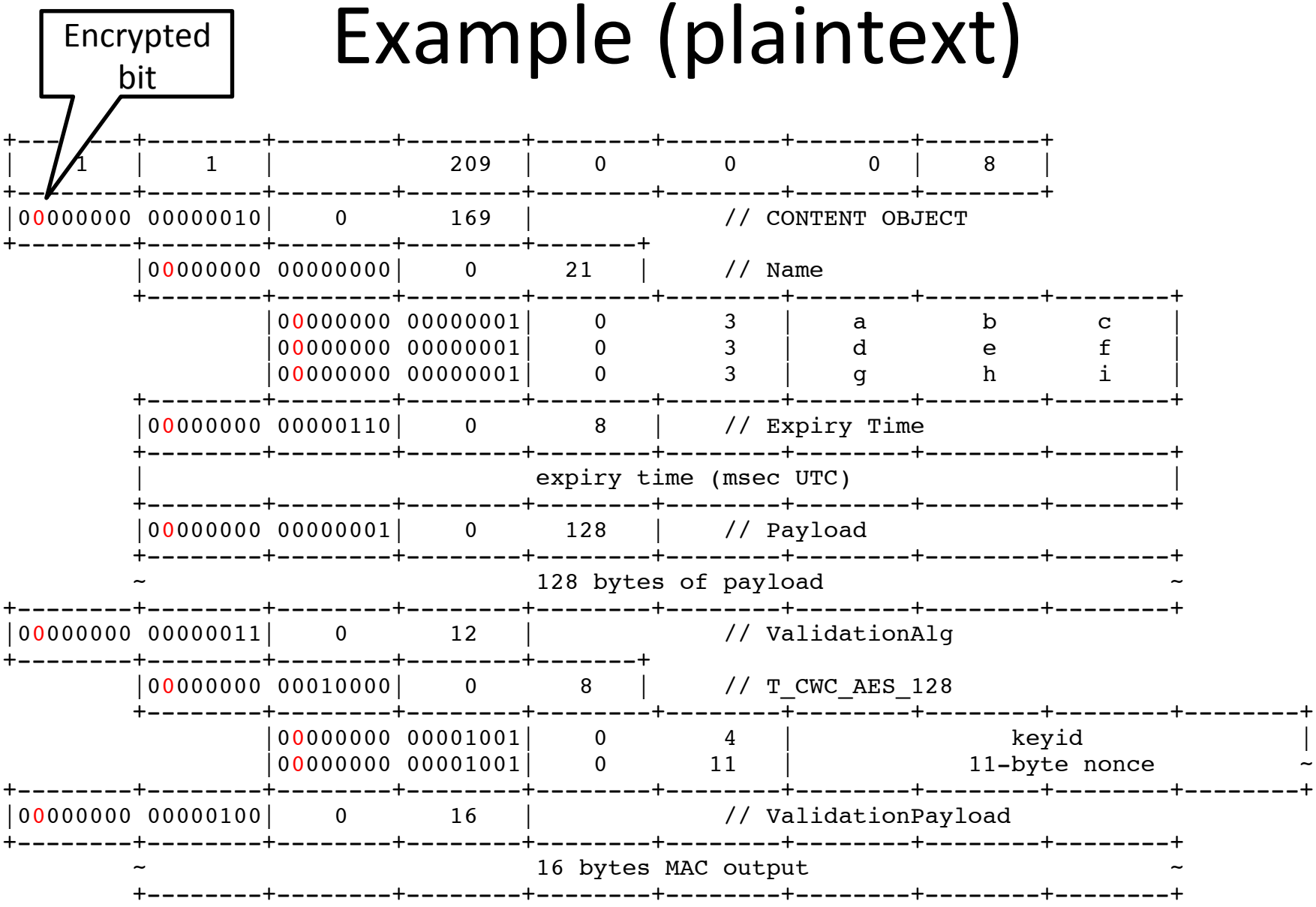
Extension

- We need to work on (A_1, M, A_2)
  - A_1 will typically be 0 bytes.
  - A_2 will be the ValidationAlg TLV section.
  - Change CWC-MAC_K(N,A,s) to CWC-MAC_K(N,A_1,s,A_2):
    - $R \leftarrow BC_K(\text{CWC-}HASH_K(A_1,s,A_2))$
  - CWC-$HASH_K$(A1,s,A2):
    - $X \leftarrow A1 \,||\, 0^{\ell} || s || 0^{\ell'} || A2 || 0^{\ell''}$
    - 0 lengths to pad to 96 bit boundary
  - Final message is (A_1, s, A_2, t)
  - S will be selectively encrypted

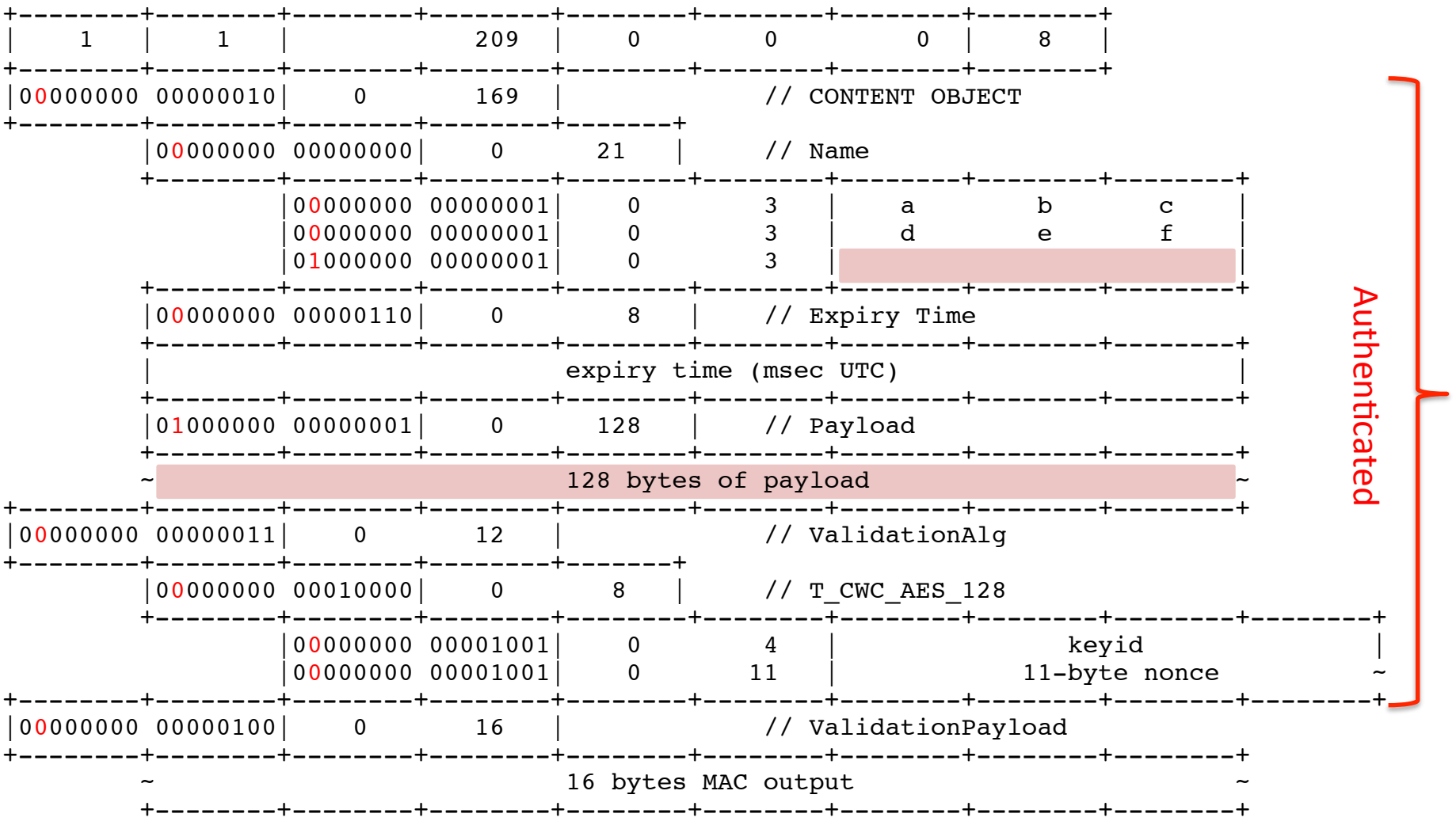
Properties

- Fast, parallelizable, AES-based.
- Simple extension of CWC for header-message-header format.
- Allows selective encryption of some metadata in the CCNx message.
- Combined authentication with encryption.

Example (plaintext)



Example (ciphertext)



Comments

- Starting cipher text at byte 0 of 173
  - Encrypt name component (3 bytes)
  - Encrypt payload (128 bytes)
  - There are 11 cipher blocks
    - Name component in middle of block 4.
    - Payload starts in middle of block 6 through block 11.
    - Pre-scanning the TLV structure means we do not need to generate 4 of the 11 AES blocks.
- One must parse the end of the packet for keyid and nonce.