# Hashing Authentication Credentials

draft-selander-lake-cred-hash-00

Göran Selander, John Preuß Mattsson, Ericsson AB
Mališa Vučinić, Inria

IETF 124, LAKE WG, Montreal, Nov 06, 2024

# Background

- EDHOC supports different authentication credentials
  - CRED can be of X.509, CWT, CCS, C509, . . .
- ... and different options for identifying credentials (by value or by reference)
  - ID_CRED can be x5chain, x5bag, x5u, x5t, c5c, c5b, c5u, c5t, kcwt, kccs, kid, . . .
- Rationale:
  - To support various IoT settings
  - To enable reduction of overhead over constrained links

- Example: ID_CRED_I containing x5u (uri of cert) and x5t (hash of cert)
  - CRED_I is not sent
  - ***But computing message_3 involves CRED_I***
  - Similar for CRED_R and message_2

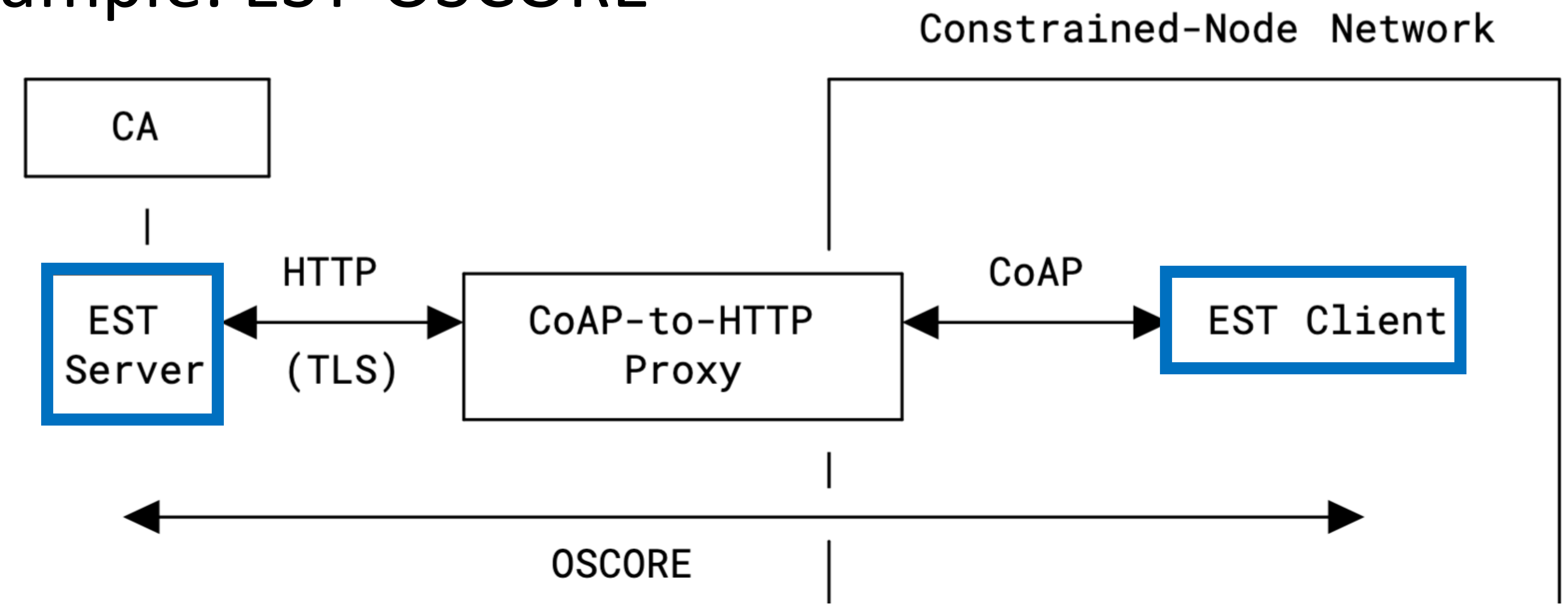
# Use of CRED in EDHOC protocol computations

- CRED_R is used in:
  - context_2 = << C_R, ID_CRED_R, TH_2, CRED_R, ? EAD_2 >>
  - external_aad = << TH_2, CRED_R, ? EAD_2 >>
  - TH_3 = H(TH_2, PLAINTEXT_2, CRED_R)

- CRED_I is used in:
  - context_3 = << ID_CRED_I, TH_3, CRED_I, ? EAD_3 >>.
  - external_aad = << TH_3, CRED_I, ? EAD_3 >>
  - TH_4 = H(TH_3, PLAINTEXT_3, CRED_I)

- So, both Initiator and Responder need CRED_R and CRED_I
- But authentication credentials are increasingly becoming very large objects
  - want to avoid unnecessary transport over constrained links

# Example: EST-OSCORE




Certificate enrolment:

POST /sen, Content-Format: application/pkcs10

2.04, Content-Format: application/cose-certhash

— **NOTE: HTTP/CoAP proxy not relevant for this discussion, it was the only figure in ace-coap-est-oscore ☺**

# Relax the need for CRED in message processing

- CRED_R can be replaced by SH(CRED_R), where **SH()** is a secure hash function

  - context_2 = << C_R, ID_CRED_R, TH_2, **SH(**CRED_R**)**, ? EAD_2 >>
  - external_aad = << TH_2, **SH(**CRED_R**)**, ? EAD_2 >>
  - TH_3 = H(TH_2, PLAINTEXT_2, **SH(**CRED_R**)**)

- To avoid breaking protocol change, we propose to signal the use of SH(CRED_R) instead of CRED_R with a **new COSE header** included in ID_CRED_R.

- Similarly, with the new COSE header in ID_CRED_I:
  - context_3 = << ID_CRED_I, TH_3, **SH(**CRED_I**)**, ? EAD_3 >>.
  - external_aad = << TH_3, **SH(**CRED_I**)**, ? EAD_3 >>
  - TH_4 = H(TH_3, PLAINTEXT_3, **SH(**CRED_I**)**

# New COSE header and hash algorithm

- Sketch in version -00 (current version):
  - New COSE header ”**Hashed Credential**”
  - No value, works like a **flag**:
  - **Replace credential with SH(credential)** in protocol computations
  - For EDHOC, SH() = H(), i.e. the hash function in the selected cipher suite

- Quick review by Marco (thanks!) included two important comments:
  - Cannot omit value in COSE header
  - Allow a different secure hash function to be used for this purpose

- Solution (planned version -01):
  - Hash algorithm identifier as value of Hashed Credential

# Summary

— We propose to define a new COSE header parameter ”Hashed Credential”, the value is the algorithm identifier of a secure hash function SH()

— When used in ID_CRED_x, CRED_x is replaced with SH(CRED_x), x = I or R

— This relaxes the need to use CRED in EDHOC processing

— For example: a device can enrol a certificate (using EST-OSCORE) and authenticate using this certificate (with EDHOC) without ever transporting the certificate over the constrained the link to the device.

— Note: security considerations are still missing, special considerations are needed for certificate metadata etc., but doesn’t seem impossible to compensate for.