EDHOC

draft-ietf-lake-edhoc-07

LAKE WG interim, June 1, 2021

Outline

- Main changes – 06 → – 07
- Selected issues

Main Changes –06 → –07

- Changed transcript hash definition for TH_2 and TH_3
- Removed "EDHOC signature algorithm curve" from cipher suite
- New application defined parameter "context" in EDHOC-Exporter
- New IANA registry "EDHOC Exporter Label"
- Moved key derivation for OSCORE to draft-ietf-core-oscore-edhoc
- Changed normative language for failure from MUST to SHOULD send error
- Made error codes non-negative and 0 for success
- Added detail on success error code
- New appendix on compact EC point representation
- Added detail on compact representation of ephemeral public keys
- Aligned terminology "protocol instance" -> "session"
- Renamed "Auxililary Data" as "External Authorization Data"
- Added encrypted EAD_4 to message_4
- Additional security considerations

Selected Issues

RPK by value

- #125 CRED_x in CWT format
- #115 Transfer CWT
- #88 Opportunistic use
- #82 COSE header map for public key
- #62 COSE_Key content constrained according to EDHOC

Correlation, message format & size

- #118 Value for C_1
- #105 Simplifying the correlation
- #103 Optimization of message size
- #61 Change message_1 format
- #39 Add guidelines for distinguishing received messages.

Conn. and key identifiers

- #79 Coding density for bstr_identifier

Inner MAC

- #121 Replace inner COSE_Encrypt0 with single invocation of EDHOC-KDF()

- #120 Initial set of cipher suites

CRED_x for non-PKI (”RPK by value”)

- EDHOC supports transport of credential in ID_CRED_x
- COSE header indicates what is being transported

ID_CRED_x = { COSE header : CRED_x }

- Solved for the PKI case: x5chain for X.509, c5c for C509

What to transport and which COSE header to use in case of RPK?

Related problem:

- What CRED_x to use in case in case RPK is **not** transported?
- Both I and R need to reproduce identical format.
- Previous version for the RPK case:
  - CRED_x an ordered subset of a COSE_key

```
CRED_x = {
  1:  1,
 -1:  4,
 -2:  h'b1a3e89460e88d3a8d54211dc95f0b90
        3ff205eb71912d6db8f4af980d2db83a',
 "subject name":"42-50-31-FF-EF-37-32-39"
}
```

Solution candidates

1. Plain COSE_key (similar to example on previous slide)
   - Define COSE header
   - Deterministic encoding
   - Label for "subject name”
2. CWT (upper example)
   - Define COSE header
   - Deterministic encoding
   - Claims list only?
3. Self-signed C509 / COSE_Sign-CWT
   - Overhead of signature
4. C509 without signature (lower example)

— New type of C509

5. Other?

```
CRED_x = {     /CWT claims list/
 2: "42-50-31-FF-EF-37-32-39",    /sub/
 8:{    /cnf/
   1:{     /COSE_Key/
     1:  1,
    -1:  4,  /X25519/
    -2:  h'b1a3e89460e88d3a8d54211dc95f0b
      903ff205eb71912d6db8f4af980d2db83a',
      }
   }
}
```

```
CRED_x = {     /C509 without signature/
2,  /new type of C509/
h'',
[],
null,
null,
h’42503137323239’, /subject name EUI-64/
1,              /P-256/
h'b1a3e89460e88d3a8d54211dc95f0b
  903ff205eb71912d6db8f4af980d2db83a',
1  /keyUsage digital signature/}
```

Correlation

- Connection identifiers in beginning of each message used for retrieving security context
- Correlation of transport messages allows connection identifiers to be omitted
  - Specified by `corr`
- *Comment: corr and optionality of connection identifiers creates complexity*

Proposal: Move message-initial connection ids from EDHOC to transport protocol & remove corr from protocol

- See PR #117
- Note: connection ids, and their negotiation, is still included for the benefit of applications

Message sizes

- Proposed changes has minor impact on message sizes
- If all changes are applied, an increase by one byte of the minimal size of one of the messages
- Acceptable?

- Recap target message sizes
  - Largest message is message_2, 46 bytes
  - Most severe restriction, 45 bytes downlink, from 6TiSCH 5-node benchmark
    - Malisa revisited the calculations and compiled a spread sheet, see #103
  - We can reach this by using known lengths
    - E.g. concatenate G_Y and CIPHERTEXT2 in one bstr
    - But that adds complexity, contrary to the latest proposed changes

- **Discuss: Tradeoff between encoding complexity and single bytes**

Compact identifiers

- bstr_identifier introduced to allow transport of short identifiers (e.g. using 1-byte CBOR ints)
- defines mapping to bytes strings that avoids collisions
- used for connection ids and transport of kids
- *Comment: Over-optimization*

Proposal: replace bstr_identifier with: bstr / int – see PR #122

- Issue: Mapping to byte strings
  - Connection ids are used as OSCORE Sender ID, need to be non-overlapping
    - So, same mapping issue but moved to draft-ietf-core-oscore-edhoc
  - COSE kid is bstr
  - If kids are transported as bstr then only one 1-byte value – empty string – can be used
    - but plenty of 2-byte values
    - Moreover, bstr_identifier only has 48 1-byte values
    - Will people really use the optimization which provides 1 byte gain in the use cases where this optimization is critical?

Simplify MAC calculation

- Current inner MACs are COSE_Encrypt0
  - message_2 and message_3

Proposal: Replace with single invocation of EDHOC-KDF()

- Improved security
- Simpler
  - "K_2m", "K_3m", "IV_2m", "IV_3m" can be removed from the specification.
- Avoids issues of erroneous use of COSE AEAD without MAC
  - Requested for FIDO alliance and other applications

- See PR #123

```
OLD
* Compute an inner COSE_Encrypt0
   * protected =  << … >>
   * external_aad = << … >>

   * plaintext = h''
   * Key K = EDHOC-KDF( …)
   * Nonce N = EDHOC-KDF( … )
   * Plaintext P = 0x

MAC_2 is the 'ciphertext’
of the inner COSE_Encrypt0.
```

```
NEW
Compute MAC_2 = EDHOC-KDF(…).
```

Cipher suites

- Is it worth having 4 different CCM based cipher suites
  - Are these the correct ones?
- Define a ChaCha20-Poly1305 cipher suite with SHA-256, X25519 and EdDSA?
- The CNSA cipher suite does not really need a 1 byte value. Change to 2 byte value?