Combining EDHOC and OSCORE

draft-palombini-core-oscore-edhoc-02

Francesca Palombini, Ericsson
Marco Tiloca, RISE
Rikard Höglund, RISE
Stefan Hristozov, Fraunhofer AISEC
Göran Selander, Ericsson

IETF 110, CoRE WG, March 8th, 2021

Recap

- Optimization for combining EDHOC (run over CoAP) with OSCORE
  - Combines EDHOC message_3 and the first subsequent OSCORE request
    - In a single EDHOC + OSCORE request, transporting both
  - Reduces the number of round trips required
    - To set up the OSCORE Security Context
    - To complete the first OSCORE transaction with that Context

- Detailed contribution
  - Method for signalling the combined message
  - Format and processing of the EDHOC + OSCORE request
  - Example of encoded EDHOC + OSCORE request

Original way: EDHOC then OSCORE

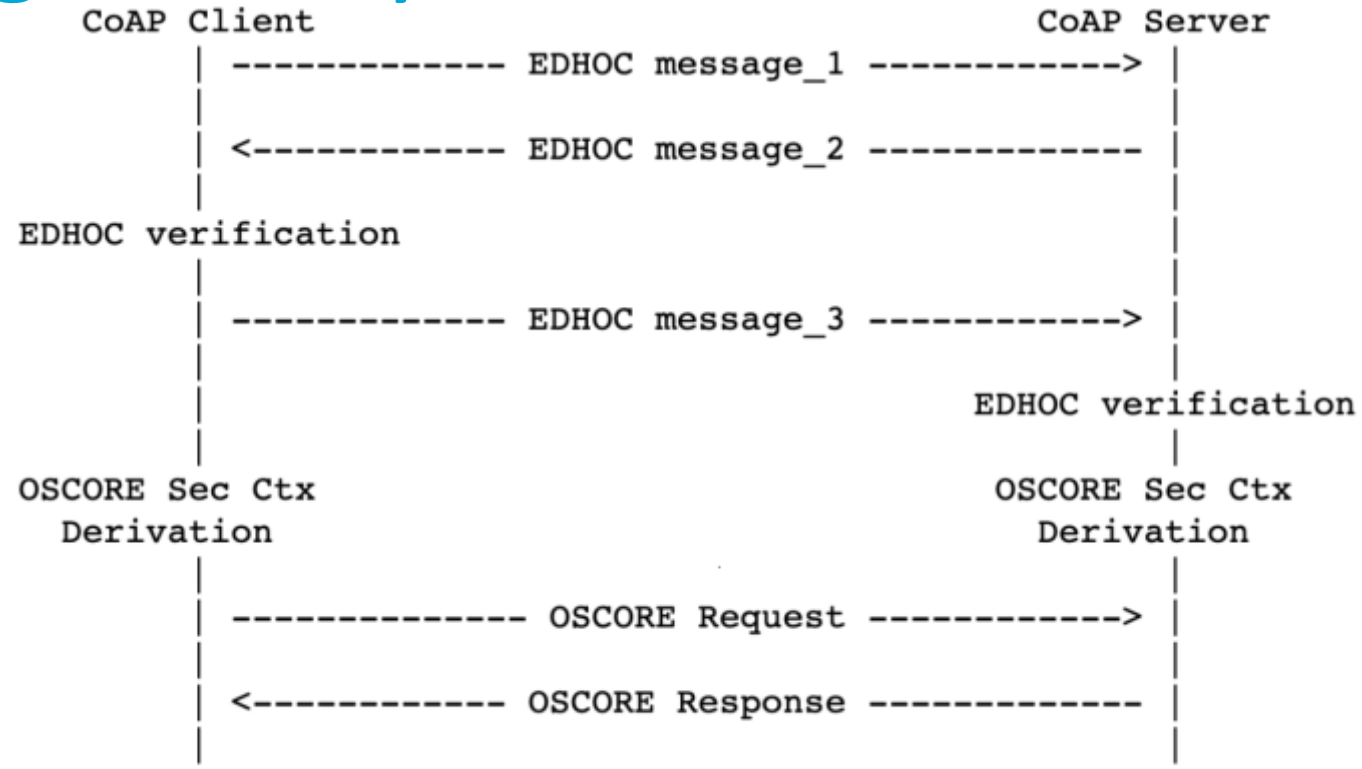


Figure 1: EDHOC and OSCORE run sequentially

New way: EDHOC + OSCORE Request

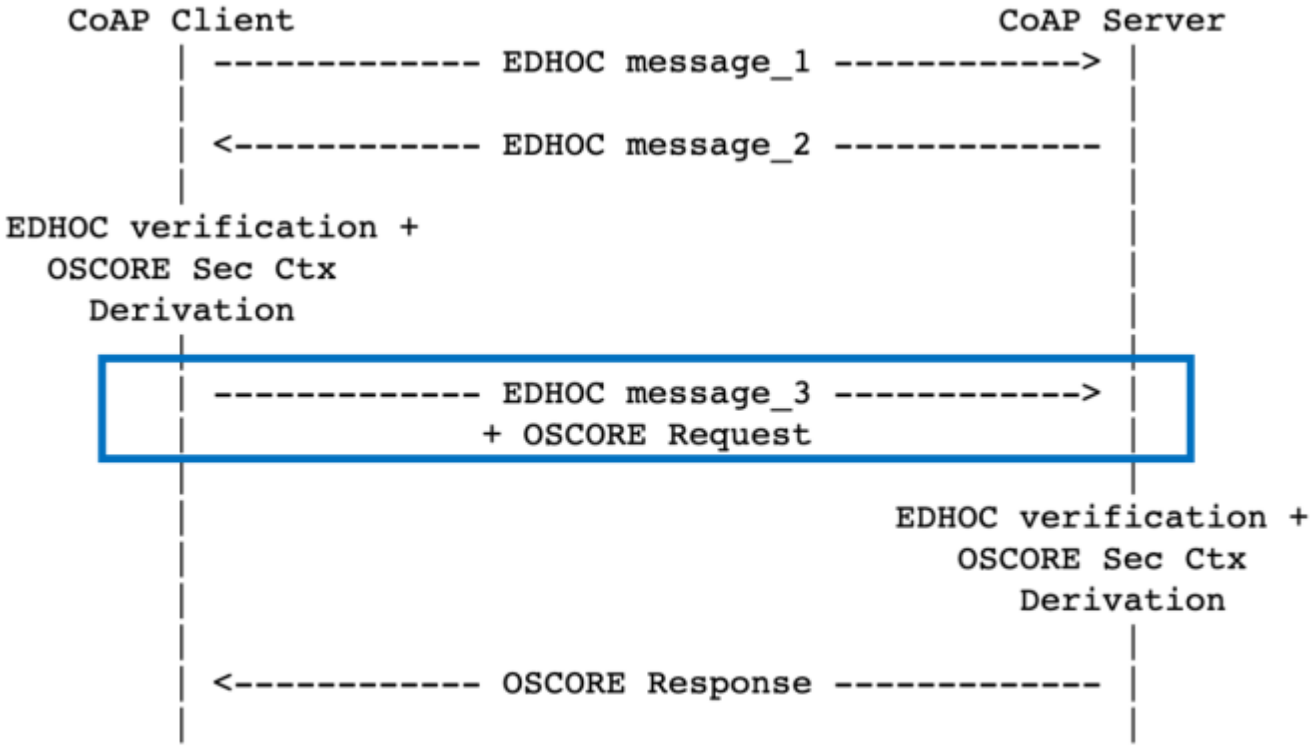


Figure 2: EDHOC and OSCORE combined

Updates from -02

- Single method for signalling the combined message
  - Use a new EDHOC option (zero-length); class U for OSCORE
  - Intended only for the EDHOC + OSCORE request
  - Based on preference from IETF 109, and feedback from implementers

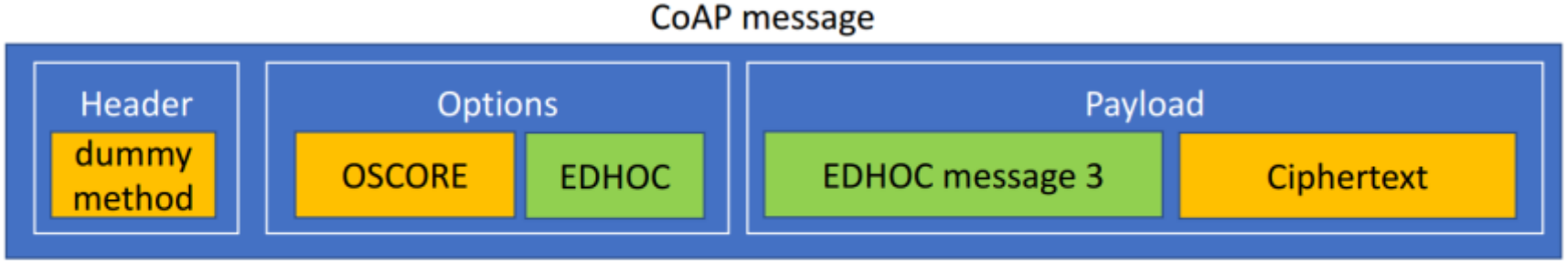
- Proposed suitable option number 13 to keep the overall option size of 1 byte
  - That's because the OSCORE option (9) is always present
  - Hence, the delta for the EDHOC option is less than 12
  - Note: option number 21 would work fine as well

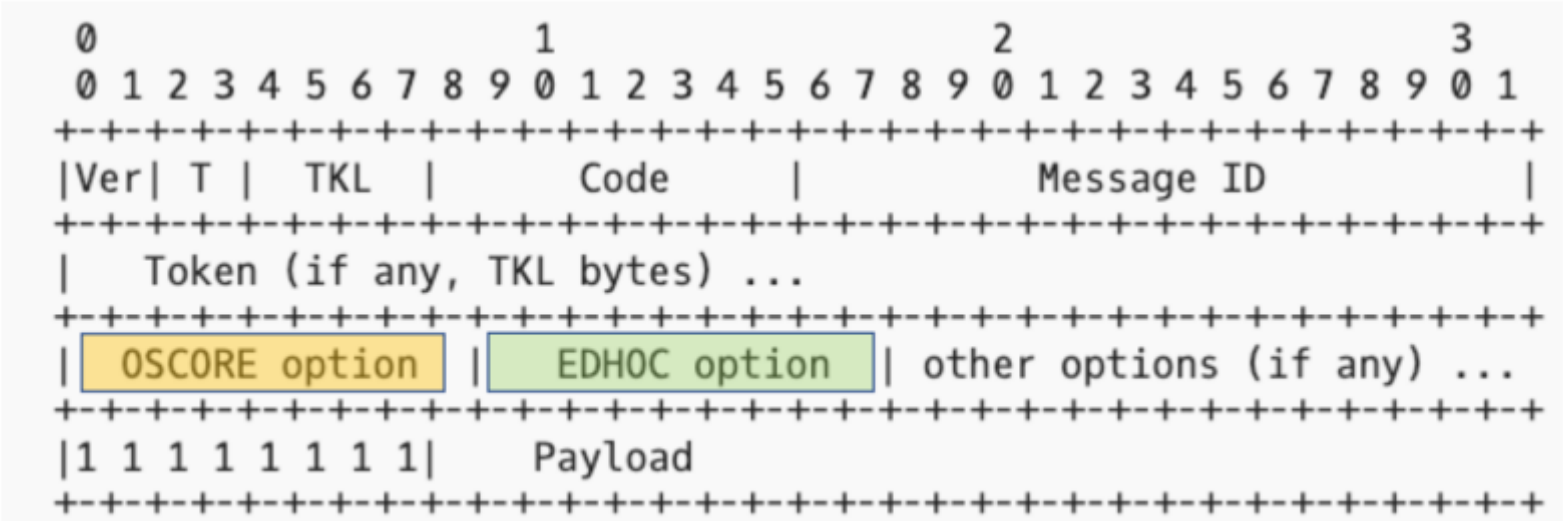
No.	C	U	N	R	Name	Format	Length	Default
TBD13	x				EDHOC	Empty	0	(none)

C=Critical, U=Unsafe, N=NoCacheKey, R=Repeatable

Figure 3: The EDHOC Option.

EDHOC + OSCORE request





Updates from -02

- Section restructuring and editorial improvement
  - Consistent with the signalling using the EDHOC option
- Improved step-by-step description of message processing
  - Detailed steps on client and server side
- Client (EDHOC Initiator):
  - Prepare EDHOC message_3 and OSCORE request; combine and send
- Server (EDHOC Responder):
  - Receive combined request; extract and process EDHOC message_3; derive OSCORE context; process the OSCORE request

Updates from -02

- Further optimization in the EDHOC + OSCORE request
  - Avoid the Sender ID of the Client to be redundant information!
    - C_R in the full EDHOC message_3 (always present in this setup)
    - 'kid' field in the OSCORE option (always present in a request)
- The combined request has a partial EDHOC message_3, that:
  - Does not include C_R
  - Includes just CIPHERTEXT_3 as a CBOR byte string
  - This saves at least 2-4 bytes on the wire
- The server rebuilds the full EDHOC message_3
  - Takes 'kid' from the OSCORE option
  - Encodes it as a bstr_identifier, as per EDHOC
  - Rebuilds the CBOR Sequence [C_R , CIPHERTEXT_3]

```
message_3 = (
  data_3,
  CIPHERTEXT_3 : bstr,
)

data_3 = (
  ? C_R : bstr_identifier,
)
```

EDHOC Message_3

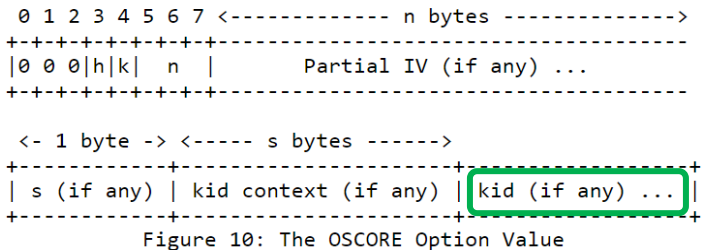


Figure 10: The OSCORE Option Value

Updates from -02

- Improved error handling on the server side
  - Details on behavior when EDHOC processing fails
  - Considerations on error code and content format to use

- EDHOC processing failure
  - Return an EDHOC Error Message
  - This will be a non-protected response to an OSCORE protected request
  - Unlike in the EDHOC draft, need to use CoAP error codes, i.e. 4.00 or 5.00
  - Use content format application/edhoc, to distinguish from OSCORE errors

- OSCORE processing failure
  - Same as in RFC 8613

Next Steps

- Keep in sync with the main EDHOC document
  - Specific points on CoAP and OSCORE may fit better in this draft
- More feedback is welcome
- WG adoption ?

Thank you!

Comments/questions?

https://github.com/EricssonResearch/oscore-edhoc