

CoRE Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: 6 May 2021

F. Palombini  
Ericsson  
M. Tiloca  
R. Hoeglund  
RISE AB  
S. Hristozov  
Fraunhofer AISEC  
G. Selander  
Ericsson  
2 November 2020

Combining EDHOC and OSCORE  
draft-palombini-core-oscore-edhoc-01

Abstract

This document defines possible optimization approaches for combining the lightweight authenticated key exchange protocol EDHOC run over CoAP with the first subsequent OSCORE transaction. This combination reduces the number of round trips required to set up an OSCORE Security Context and complete an OSCORE transaction using that context.

Discussion Venues

This note is to be removed before publishing as an RFC.

Source for this draft and an issue tracker can be found at  
<https://github.com/EricssonResearch/oscore-edhoc>  
(<https://github.com/EricssonResearch/oscore-edhoc>).

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 6 May 2021.

## Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
1.1. Terminology . . . . .	3
2. Background . . . . .	3
3. EDHOC in OSCORE . . . . .	5
3.1. Signalling in a New EDHOC Option . . . . .	6
3.2. Signalling in the OSCORE Option . . . . .	8
4. Security Considerations . . . . .	9
5. IANA Considerations . . . . .	9
6. Normative References . . . . .	9
Acknowledgments . . . . .	10
Authors' Addresses . . . . .	10

## 1. Introduction

This document presents possible optimization approaches to combine the lightweight authenticated key exchange protocol EDHOC [I-D.ietf-lake-edhoc], when running over CoAP [RFC7252], with the first subsequent OSCORE [RFC8613] transaction.

This allows for a minimum number of round trips necessary to setup the OSCORE Security Context and complete an OSCORE transaction, for example when an IoT device gets configured in a network for the first time.

The number of protocol round trips impacts the minimum number of flights, which can have a substantial impact on performance with certain radio technologies.

Without this optimization, it is not possible, not even in theory, to achieve the minimum number of flights. This optimization makes it possible also in practice, since the last message of the EDHOC protocol can be made relatively small (see Section 1 of [I-D.ietf-lake-edhoc]), thus allowing additional OSCORE protected CoAP data within target MTU sizes.

The goal of this document is to provide details on different alternatives for transporting and processing the necessary data, gather opinions on the different approaches, and select only one of those.

### 1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

The reader is expected to be familiar with terms and concepts defined in CoAP [RFC7252], CBOR [I-D.ietf-cbor-7049bis], OSCORE [RFC8613] and EDHOC [I-D.ietf-lake-edhoc].

## 2. Background

EDHOC is a 3-message key exchange protocol. Section 7.1 of [I-D.ietf-lake-edhoc] specifies how to transport EDHOC over CoAP: the EDHOC data (referred to as "EDHOC messages") are transported in the payload of CoAP requests and responses.

This draft deals with the case of the Initiator acting as CoAP Client and the Responder acting as CoAP Server. (The case of the Initiator acting as CoAP server cannot be optimized in this way.) That is, the CoAP Client sends a POST request containing the EDHOC message 1 to a reserved resource at the CoAP Server. This triggers the EDHOC exchange on the CoAP Server, which replies with a 2.04 (Changed) Response containing the EDHOC message 2. Finally, the EDHOC message 3 is sent by the CoAP Client in a CoAP POST request to the same resource used for the EDHOC message 1. The Content-Format of these CoAP messages is set to "application/edhoc".

After this exchange takes place, and after successful verifications specified in the EDHOC protocol, the Client and Server derive the OSCORE Security Context, as specified in Section 7.1.1 of [I-D.ietf-lake-edhoc]. Then, they are ready to use OSCORE.

This sequential way of running EDHOC and then OSCORE is specified in Figure 1. As shown in the figure, this mechanism is executed in 3 round trips.

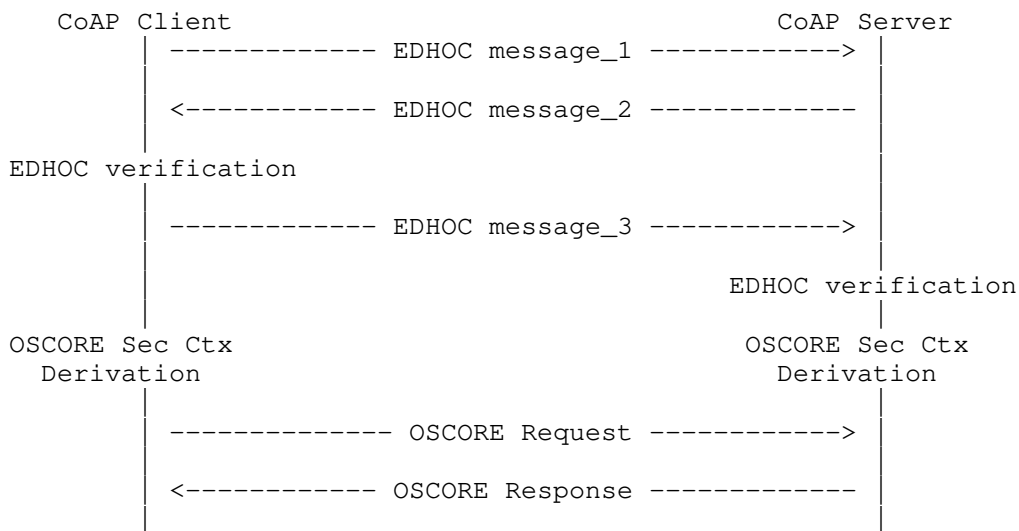


Figure 1: EDHOC and OSCORE run sequentially

The number of roundtrips can be minimized: after receiving the EDHOC message 2, the CoAP Client has all the information needed to derive the OSCORE Security Context before sending the EDHOC message 3.

This means that the Client can potentially send at the same time both the EDHOC message 3 and the subsequent OSCORE Request. On a semantic level, this approach practically requires to send two separate REST requests at the same time.

The high level message flow of running EDHOC and OSCORE combined is shown in Figure 2.

Defining the specific details of how to transport the data and of their processing order is the goal of this specification.

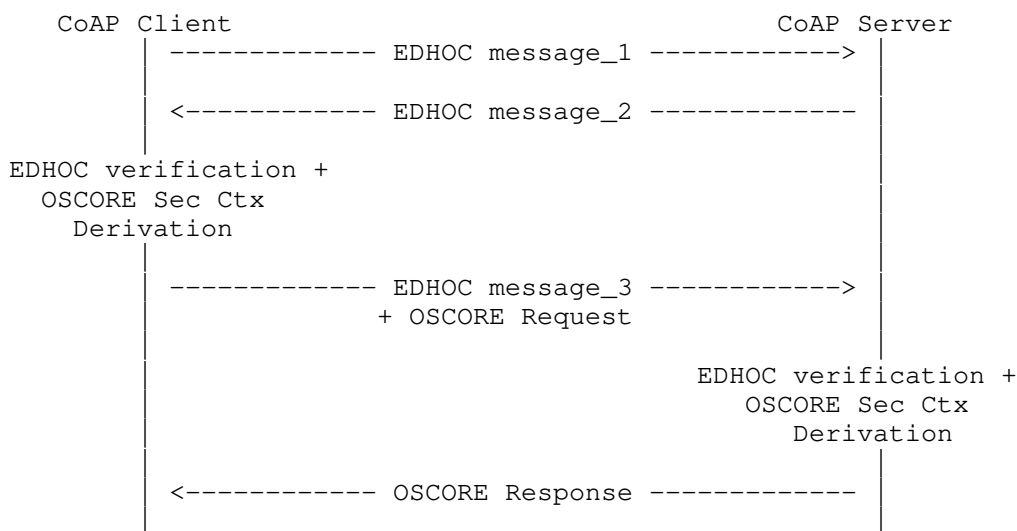


Figure 2: EDHOC and OSCORE combined

### 3. EDHOC in OSCORE

This approach consists in sending the EDHOC message 3 inside an OSCORE message (i.e., an OSCORE protected CoAP message).

The resulting OSCORE + EDHOC request is in practice the OSCORE Request from Figure 1, sent to a protected resource and with the correct CoAP method and options, with the addition that it also transports the EDHOC message 3.

As the EDHOC message 3 may be too large to be included in a CoAP Option, e.g. if containing a large public key certificate chain, it would have to be transported in the CoAP payload.

In particular, the payload of the OSCORE + EDHOC request is formatted as a CBOR sequence of two CBOR byte strings: the EDHOC message 3 and the OSCORE ciphertext of the original OSCORE Request, in this order, both encoded as CBOR byte strings.

Note that the OSCORE ciphertext is not computed over the EDHOC message 3, which is not protected by OSCORE. That is, the client first prepares the OSCORE Request as in Figure 1. Then, it reformats the payload to include also the EDHOC message 3, as defined above. The result is the OSCORE + EDHOC request to send.

The usage of this approach is indicated by a signalling information in the OSCORE + EDHOC request, which can be either a new EDHOC Option (see Section 3.1) or the OSCORE Option with a particular Flag Bit set (see Section 3.2).

When receiving such a request, the Server needs to perform the following processing, in addition to the EDHOC, OSCORE and CoAP processing:

1. Check the signalling information to identify that this is an OSCORE + EDHOC request.
2. Extract the EDHOC message 3 from the payload of the OSCORE + EDHOC request, as the value of the first CBOR byte string in the CBOR sequence.
3. Execute the EDHOC processing on the EDHOC message 3, including verifications and the OSCORE Security Context derivation.
4. Extract the OSCORE ciphertext from the payload of the OSCORE + EDHOC request, as the value of the second CBOR byte string in the CBOR sequence. Then, set the CoAP payload of the request to the extracted ciphertext.
5. Decrypt and verify the OSCORE protected CoAP request resulting from step 4, as defined by OSCORE.
6. Process the CoAP request resulting from step 5.

The following sections expand on the two ways of signalling that the EDHOC message is transported in the OSCORE message.

### 3.1. Signalling in a New EDHOC Option

One way to signal that the Server has to extract and process the EDHOC message 3 before processing the OSCORE protected CoAP request is to define a new CoAP Option, called the EDHOC Option.

The presence of this option means that the message contains EDHOC data in the payload, that must be extracted and processed before the rest of the message can be processed.

In particular, the EDHOC message 3 has to be extracted from the CoAP payload, as the first element of a CBOR sequence wrapped in a CBOR byte string.

The Option is critical, Safe-to-Forward, and part of the Cache-Key.

The Option value is always empty. If any value is sent, the value is simply ignored.

The Option MUST occur at most once.

The Option is of Class U for OSCORE.

Figure 3 shows the format for a CoAP message containing both the OSCORE ciphertext and EDHOC message 3, using the newly defined EDHOC option for signaling.

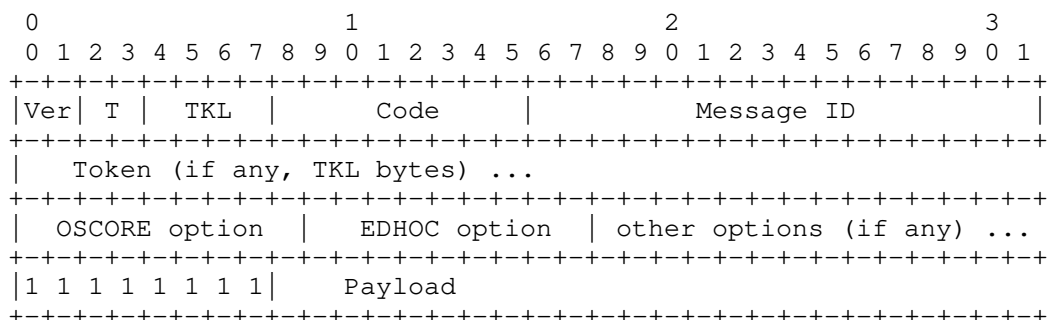


Figure 3: CoAP message for EDHOC and OSCORE combined - signaled with the EDHOC Option

An example based on the OSCORE test vector from Appendix C.4 of [RFC8613] and the EDHOC test vector from Appendix B.2 of [I-D.ietf-lake-edhoc] is given in Figure 4. The example assumes that the EDHOC option is registered with CoAP option number 13.

- o OSCORE option value: 0x0914 (2 bytes)
- o ciphertext: 0x612f1092f1776f1c1668b3825e (13 bytes)
- o EDHOC option value: - (0 bytes)
- o EDHOC message 3: 085253c3991999a5ffb86921e99b607c067770e0 (20 bytes)

From there:

- o Protected CoAP request (OSCORE message): 0x44025d1f00003974396c6f63616c686f737462 0914 04 ff 54085253c3991999a5ffb86921e99b607c067770e0 4d612f1092f1776f1c1668b3825e (58 bytes)

Figure 4: CoAP message for EDHOC and OSCORE combined - signaled with the EDHOC Option

### 3.2. Signalling in the OSCORE Option

Another way to signal that the EDHOC message 3 is to be extracted from the CoAP payload as the first element of a CBOR sequence wrapped in a CBOR byte string, and that the processing defined in Section 3 is to be executed, is to use one of the OSCORE Flag Bits of the OSCORE Option.

Bit Position: 1

Name: EDHOC

Description: Set to 1 if the payload is a sequence of EDHOC message 3 and OSCORE ciphertext.

Reference: this document

The OSCORE Option value with the EDHOC bit set is given in Figure 5.

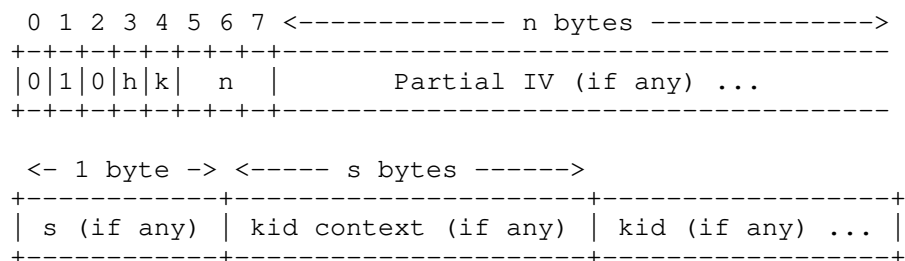


Figure 5: The OSCORE Option Value with the EDHOC bit set

Figure 6 shows the format for a CoAP message containing both the OSCORE ciphertext and EDHOC message 3, using the Flag Bit 1 in the OSCORE Option for signaling.

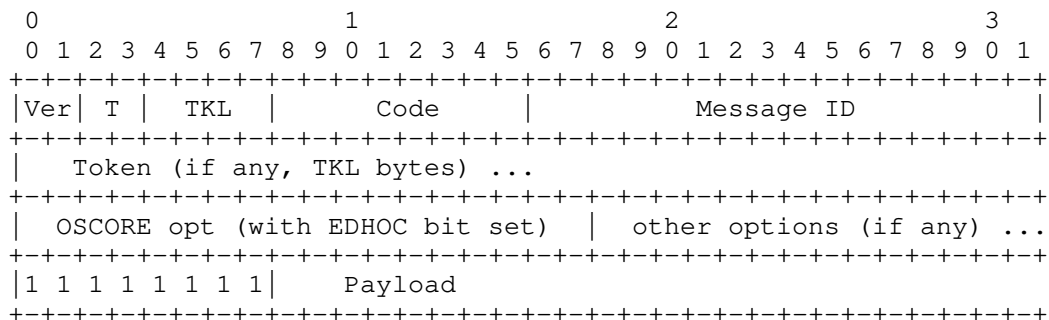




Figure 6: CoAP message for EDHOC and OSCORE combined - signaled within the OSCORE option

An example based on the OSCORE test vector from Appendix C.4 of [RFC8613] and the EDHOC test vector from Appendix B.2 of [I-D.ietf-lake-edhoc] is given in Figure 7.

- o OSCORE option value without EDHOC bit set: 0x0914 (2 bytes)
- o OSCORE option value with EDHOC bit set: 0x4914 (2 bytes)
- o ciphertext: 0x612f1092f1776f1c1668b3825e (13 bytes)
- o EDHOC message 3: 085253c3991999a5ffb86921e99b607c067770e0 (20 bytes)

From there:

- o Protected CoAP request (OSCORE message): 0x44025d1f00003974396c6f63616c686f737462 4914 ff 54085253c3991999a5ffb86921e99b607c067770e0 4d612f1092f1776f1c1668b3825e (58 bytes)

Figure 7: CoAP message for EDHOC and OSCORE combined - signaled within the OSCORE Option

#### 4. Security Considerations

The same security considerations from OSCORE [RFC8613] and EDHOC [I-D.ietf-lake-edhoc] hold for this document.

TODO (more considerations)

#### 5. IANA Considerations

Depending on the option chosen, this document will either register a new CoAP Option number to the CoAP Option Number registry, or a new bit to the OSCORE Flag Bits registry.

#### 6. Normative References

[I-D.ietf-cbor-7049bis]

Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", Work in Progress, Internet-Draft, draft-ietf-cbor-7049bis-16, 30 September 2020, <<http://www.ietf.org/internet-drafts/draft-ietf-cbor-7049bis-16.txt>>.

- [I-D.ietf-lake-edhoc]  
Selander, G., Mattsson, J., and F. Palombini, "Ephemeral Diffie-Hellman Over COSE (EDHOC)", Work in Progress, Internet-Draft, draft-ietf-lake-edhoc-01, 2 August 2020, <<http://www.ietf.org/internet-drafts/draft-ietf-lake-edhoc-01.txt>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, DOI 10.17487/RFC7252, June 2014, <<https://www.rfc-editor.org/info/rfc7252>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8613] Selander, G., Mattsson, J., Palombini, F., and L. Seitz, "Object Security for Constrained RESTful Environments (OSCORE)", RFC 8613, DOI 10.17487/RFC8613, July 2019, <<https://www.rfc-editor.org/info/rfc8613>>.

#### Acknowledgments

The authors sincerely thank Christian Amsuess, Klaus Hartke, Jim Schaad and Malisa Vucinic for their feedback and comments in the discussion leading up to this draft.

The work on this document has been partly supported by VINNOVA and the Celtic-Next project CRITISEC; and by the H2020 project SIFIS-Home (Grant agreement 952652).

#### Authors' Addresses

Francesca Palombini  
Ericsson

Email: [francesca.palombini@ericsson.com](mailto:francesca.palombini@ericsson.com)

Marco Tiloca  
RISE AB  
Isafjordsgatan 22  
SE-16440 Stockholm Kista  
Sweden

Email: marco.tiloca@ri.se

Rikard Hoeglund  
RISE AB  
Isafjordsgatan 22  
SE-16440 Stockholm Kista  
Sweden

Email: rikard.hoglund@ri.se

Stefan Hristozov  
Fraunhofer AISEC

Email: stefan.hristozov@aisec.fraunhofer.de

Goeran Selander  
Ericsson

Email: goran.selander@ericsson.com