CoRE Working Group
Internet-Draft
Intended status: Standards Track
Expires: 14 January 2021

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

13 July 2020

Combining EDHOC and OSCORE
draft-palombini-core-oscore-edhoc-00

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 14 January 2021.

Table of Contents

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 as discussed in Section 2.11 of [I-D.ietf-lake-reqs].

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 [I-D.ietf-lake-reqs].

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.

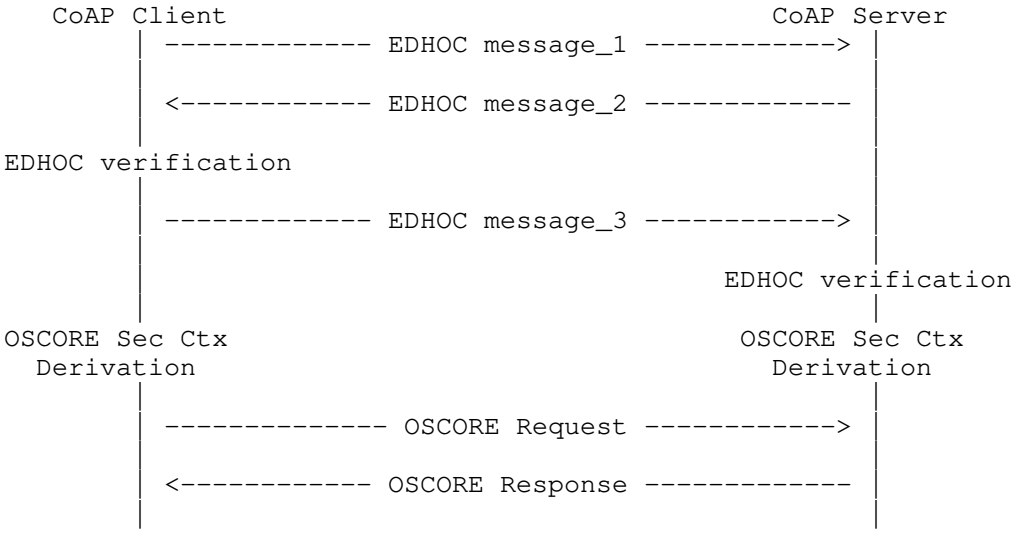

Figure 1: EDHOC and OSCORE run sequentially

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.

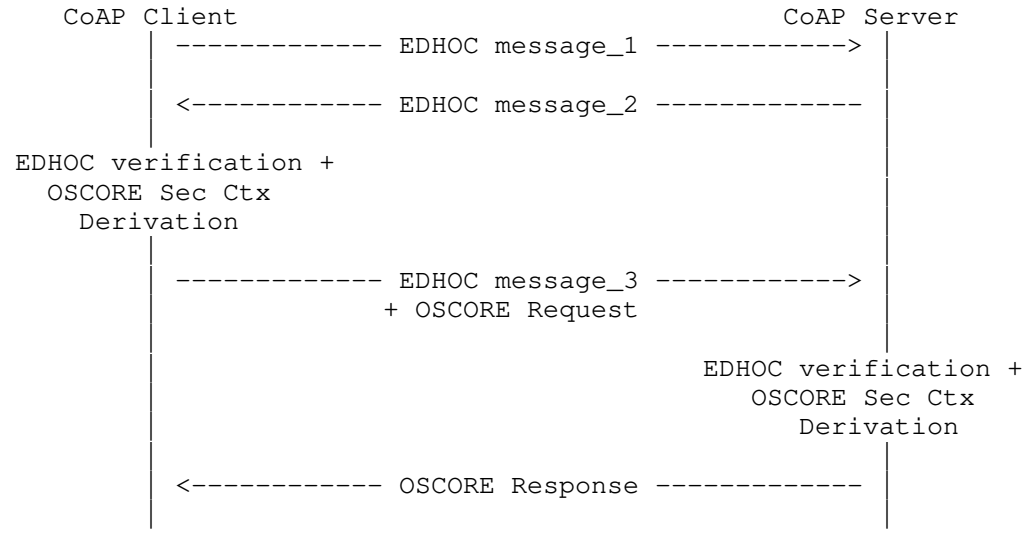

Figure 2: EDHOC and OSCORE combined

3. EDHOC in OSCORE

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

The 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.

The payload of the request is formatted as a CBOR sequence [I-D.ietf-lake-reqs] of two CBOR wrapped items: the EDHOC message 3 and the OSCORE ciphertext, in this order.

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 usage of this approach is indicated by a signalling information, 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.

3. Execute the EDHOC processing, including verifications and OSCORE Security Context derivation.

4. Decrypt and verify the remaining OSCORE protected CoAP request as defined by OSCORE.

5. Process the CoAP request.

The following sections expand on the 2 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 is to extract and process the EDHOC message 3 before the OSCORE message is processed is to define a new CoAP Option, called the EDHOC Option.

This Option being present 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 is to be extracted from the CoAP payload, as the CBOR wrapped first element of a CBOR sequence.

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 discarded.

The Option must occur at most once.

The Option is of Class U for OSCORE.

3.2. Signalling in the OSCORE Option

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

Bit Position: 8

Name: EDHOC

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

Reference: this document

4. OSCORE in EDHOC

Instead of transporting the EDHOC message inside an OSCORE message, the second approach consists in transporting the OSCORE protected data in an EDHOC message.

The request is in practice the CoAP POST Request containing the EDHOC message 3 from Figure 1, sent to the unprotected resource reserved to EDHOC processing, with the addition that it also transports the OSCORE Option and ciphertext of the original OSCORE Request.

The OSCORE Option and ciphertext contain all the information to reconstruct the original OSCORE Request, including CoAP method, options and payload.

The payload is formatted as a CBOR sequence of three CBOR wrapped items: the EDHOC message 3, the OSCORE Option and the OSCORE ciphertext, in this order.

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 protected CoAP Request. Then, it adds the OSCORE option and ciphertext to the payload of the EDHOC request to send, as defined above.

The usage of this approach is indicated by a signalling information, which can be either a new EDHOC+OSCORE option (see Section 4.1) or the particular structure of the request payload (see Section 4.2).

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

1. Check the signalling information to identify that this is an EDHOC + OSCORE request.

2. Extract the EDHOC message 3 from the payload.

3. Execute the EDHOC processing, including verifications and OSCORE Security Context derivation.

4. Extract the OSCORE Option value and ciphertext from the payload and reconstruct the OSCORE protected CoAP Request.

5. Decrypt and verify the reconstructed OSCORE protected CoAP request as defined by OSCORE.

6. Process the CoAP request.

Compared to the approach in Section 3, this processing requires one more step, as the Server must build the OSCORE protected CoAP request from the payload before being able to process it.

4.1. Signalling in a New EDHOC+OSCORE Option

One way to signal that the Server is to build and process the OSCORE protected CoAP request after the EDHOC processing is to define a new CoAP Option, called the EDHOC+OSCORE Option.

This Option being present (either in a request or response) means that the message contains an OSCORE option value and ciphertext in the payload, that must be extracted and processed after the EDHOC processing.

The OSCORE option and ciphertext are to be extracted from the CoAP payload as the CBOR wrapped second and third element of a CBOR sequence.

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 discarded.

The Option must occur at most once.

The Option is of Class U for OSCORE.

4.2. Signalling Based on the Number of Elements in the Payload

Another way to signal this approach and to mandate that the Server is to build and process the OSCORE protected CoAP request after the EDHOC processing is to set up pre-determined policies on both the Client and Server.

A Client may be set up to support at the same time receiving only the EDHOC message 3 or both the EDHOC message 3 and the OSCORE Option and ciphertext in the request. The Client would be able to distinguish the two cases based on the number of CBOR elements in the payload, and process the message accordingly.

5. Security Considerations

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

TODO (more considerations)

6. IANA Considerations

This document has no IANA actions.

7. 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-14, 16 June 2020, <http://www.ietf.org/internet-drafts/draft-ietf-cbor-7049bis-14.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-00, 6 July 2020, <http://www.ietf.org/internet-drafts/draft-ietf-lake-edhoc-00.txt>.

[I-D.ietf-lake-reqs]
Vucinic, M., Selander, G., Mattsson, J., and D. Garcia-Carillo, "Requirements for a Lightweight AKE for OSCORE", Work in Progress, Internet-Draft, draft-ietf-lake-reqs-04, 8 June 2020, <http://www.ietf.org/internet-drafts/draft-ietf-lake-reqs-04.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.

Authors' Addresses

Francesca Palombini
Ericsson

Email: francesca.palombini@ericsson.com

Marco Tiloca
RISE AB

Email: marco.tiloca@ri.se

Rikard Hoeglund
RISE AB

Email: rikard.hoglund@ri.se

Stefan Hristozov
Fraunhofer AISEC

Email: stefan.hristozov@aisec.fraunhofer.de

Goeran Selander
Ericsson

Email: goran.selander@ericsson.com