Workgroup: CoRE Working Group
Internet-Draft:
draft-ietf-core-oscore-edhoc-07
Published: 13 March 2023
Intended Status: Standards Track
Expires: 14 September 2023
Authors: F. Palombini M. Tiloca R. Hoeglund
Ericsson RISE AB RISE AB
S. Hristozov G. Selander
Fraunhofer AISEC Ericsson

Using EDHOC with CoAP and OSCORE

Abstract

The lightweight authenticated key exchange protocol EDHOC can be run over CoAP and used by two peers to establish an OSCORE Security Context. This document details this use of the EDHOC protocol, by specifying a number of additional and optional mechanisms. These especially include an optimization approach for combining the execution of EDHOC with the first OSCORE transaction. This combination reduces the number of round trips required to set up an OSCORE Security Context and to complete an OSCORE transaction using that Security Context.

Discussion Venues

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

Discussion of this document takes place on the Constrained RESTful Environments Working Group mailing list (core@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/core/.

Source for this draft and an issue tracker can be found at https://github.com/core-wg/oscore-edhoc.

Status of This Memo

This Internet-Draft will expire on 14 September 2023.

Copyright Notice

Table of Contents

1. Introduction

Ephemeral Diffie-Hellman Over COSE (EDHOC) [I-D.ietf-lake-edhoc] is a lightweight authenticated key exchange protocol, especially intended for use in constrained scenarios. In particular, EDHOC messages can be transported over the Constrained Application Protocol (CoAP) [RFC7252] and used for establishing a Security Context for Object Security for Constrained RESTful Environments (OSCORE) [RFC8613].

This document details this use of the EDHOC protocol, and specifies a number of additional and optional mechanisms. These especially include an optimization approach, that combines the EDHOC execution with the first OSCORE transaction (see Section 3). This allows for a minimum number of round trips necessary to setup the OSCORE Security Context and complete an OSCORE transaction, e.g., when an IoT device gets configured in a network for the first time.

This optimization is desirable, since the number of protocol round trips impacts on the minimum number of flights, which in turn can have a substantial impact on the latency of conveying the first OSCORE request, when using 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 1.2 of [I-D.ietf-lake-edhoc]), thus allowing additional OSCORE-protected CoAP data within target MTU sizes.

Furthermore, this document defines a number of parameters corresponding to different information elements of an EDHOC application profile (see Section 6). These can be specified as target attributes in the link to an EDHOC resource associated with that application profile, thus enabling an enhanced discovery of such resource for CoAP clients.

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 [RFC8949], OSCORE [RFC8613], and EDHOC [I-D.ietf-lake-edhoc].

2. EDHOC Overview

This section is not normative and summarizes what is specified in [I-D.ietf-lake-edhoc], in particular its Appendix A.2. Thus, it provides a baseline for the enhancements in the subsequent sections.

The EDHOC protocol specified in [I-D.ietf-lake-edhoc] allows two peers to agree on a cryptographic secret, in a mutually-authenticated way and by using Diffie-Hellman ephemeral keys to achieve forward secrecy. The two peers are denoted as Initiator and Responder, as the one sending or receiving the initial EDHOC message_1, respectively.

After successful processing of EDHOC message_3, both peers agree on a cryptographic secret that can be used to derive further security material, and especially to establish an OSCORE Security Context [RFC8613]. The Responder can also send an optional EDHOC message_4 to achieve key confirmation, e.g., in deployments where no protected application message is sent from the Responder to the Initiator.

Appendix A.2 of [I-D.ietf-lake-edhoc] specifies how to transfer EDHOC over CoAP. That is, the EDHOC data (referred to as "EDHOC messages") are transported in the payload of CoAP requests and responses. The default, forward message flow of EDHOC consists in the CoAP client acting as Initiator and the CoAP server acting as Responder. Alternatively, the two roles can be reversed, as per the reverse message flow of EDHOC. In the rest of this document, EDHOC messages are considered to be transferred over CoAP.

Figure 1 shows a CoAP client and a CoAP server running EDHOC as Initiator and Responder, respectively. That is, the client sends a POST request to a reserved EDHOC resource at the server, by default at the Uri-Path "/.well-known/edhoc". The request payload consists of the CBOR simple value "true" (0xf5) concatenated with EDHOC message_1, which also includes the EDHOC connection identifier C_I of the client encoded as per Section 3.3 of [I-D.ietf-lake-edhoc]. The Content-Format of the request can be set to application/cid-edhoc+cbor-seq.

This triggers the EDHOC execution at the server, which replies with a 2.04 (Changed) response. The response payload consists of EDHOC message_2, which also includes the EDHOC connection identifier C_R of the server encoded as per Section 3.3 of [I-D.ietf-lake-edhoc].

The Content-Format of the response can be set to application/edhoc+cbor-seq.

Finally, the client sends a POST request to the same EDHOC resource used earlier to send EDHOC message_1. The request payload consists of the EDHOC connection identifier C_R encoded as per Section 3.3 of [I-D.ietf-lake-edhoc], concatenated with EDHOC message_3. The Content-Format of the request can be set to application/cid-edhoc+cbor-seq.

After this exchange takes place, and after successful verifications as specified in the EDHOC protocol, the client and server can derive an OSCORE Security Context, as defined in Appendix A.1 of [I-D.ietf-lake-edhoc]. After that, they can use OSCORE to protect their communications as per [RFC8613].

The client and server are required to agree in advance on certain information and parameters describing how they should use EDHOC. These are specified in an application profile associated with the used EDHOC resource (see Section 3.9 of [I-D.ietf-lake-edhoc].

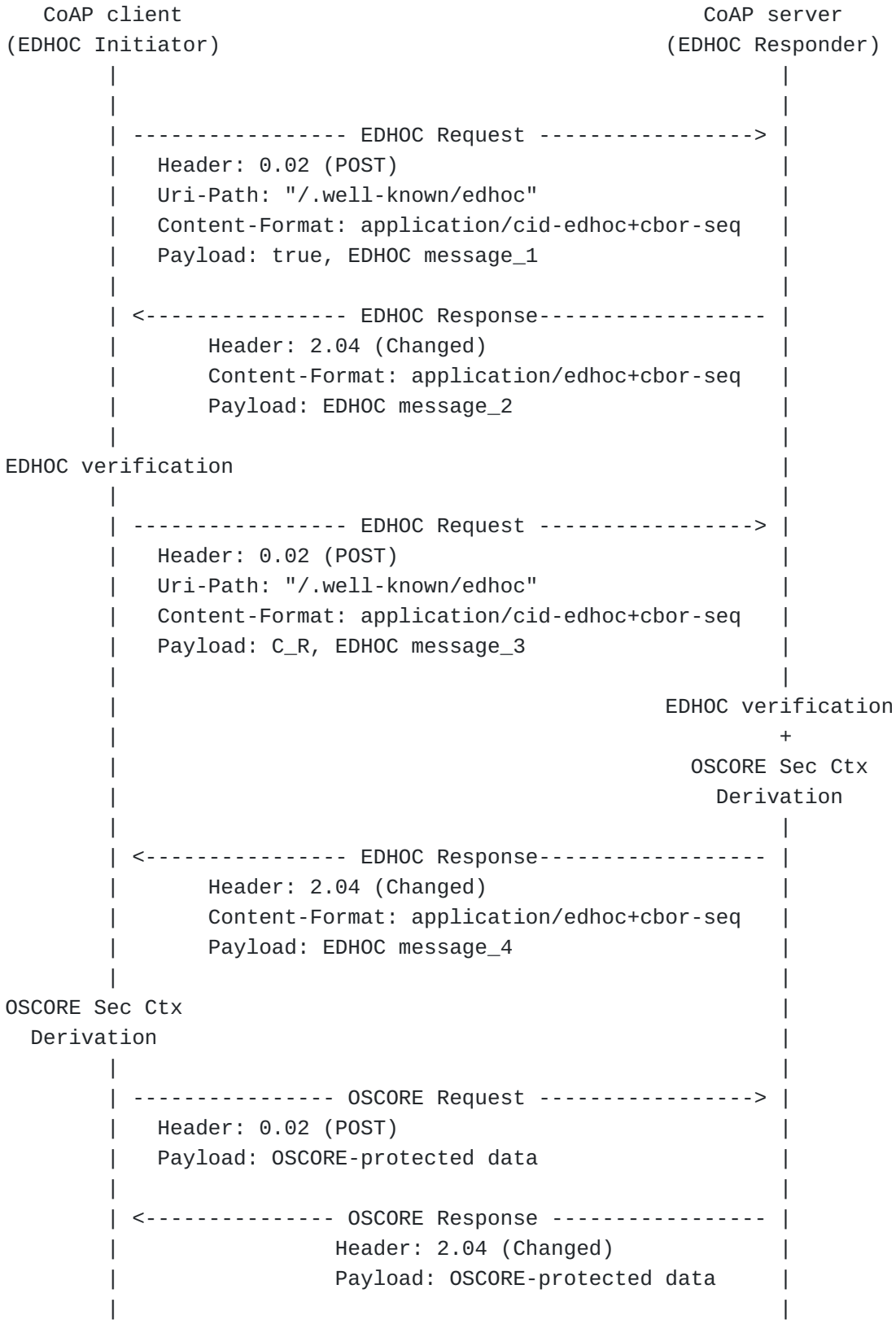

Figure 1: EDHOC and OSCORE run sequentially. The optional message_4 is included in this example, without which that message needs no payload.

As shown in Figure 1, this purely-sequential flow where EDHOC is run first and then OSCORE is used takes three round trips to complete.

Section 3 defines an optimization for combining EDHOC with the first OSCORE transaction. This reduces the number of round trips required to set up an OSCORE Security Context and to complete an OSCORE transaction using that Security Context.

3. EDHOC Combined with OSCORE

This section defines an optimization for combining the EDHOC message exchange with the first OSCORE transaction, thus minimizing the number of round trips between the two peers.

This approach can be used only if the default, forward message flow of EDHOC is used, i.e., when the client acts as Initiator and the server acts as Responder. That is, it cannot be used in the case with reversed roles as per the reverse message flow of EDHOC.

When running the purely-sequential flow of Section 2, the client has all the information to derive the OSCORE Security Context already after receiving EDHOC message_2 and before sending EDHOC message_3.

Hence, the client can potentially send both EDHOC message_3 and the subsequent OSCORE Request at the same time. On a semantic level, this requires sending two REST requests at once, as in Figure 2.

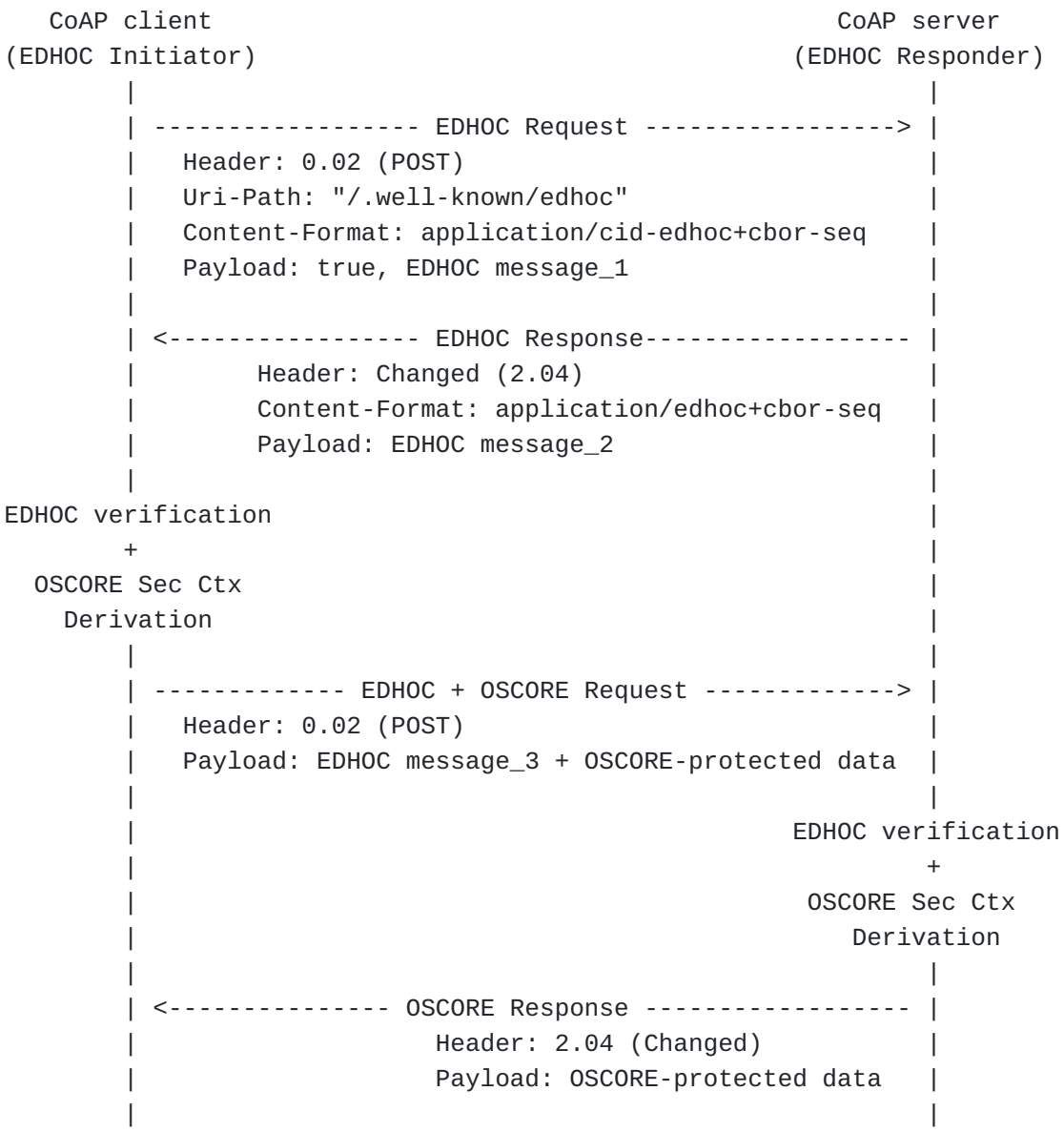

Figure 2: EDHOC and OSCORE combined.

To this end, the specific approach defined in this section consists of sending a single EDHOC + OSCORE request, which conveys the pair (C_R, EDHOC message_3) within an OSCORE-protected CoAP message.

That is, the EDHOC + OSCORE request is in practice the OSCORE Request from Figure 1, as still sent to a protected resource and with the correct CoAP method and options intended for accessing that resource. At the same time, the EDHOC + OSCORE request also transports the pair (C_R, EDHOC message_3) required for completing the EDHOC session. Note that, as specified in Section 3.2, C_R is transported in the OSCORE Option rather than in the request payload.

Since EDHOC message_3 may be too large to be included in a CoAP Option, e.g., if conveying a protected large public key certificate

chain as ID_CRED_I (see Section 3.5.3 of [I-D.ietf-lake-edhoc]) or if conveying protected External Authorization Data as EAD_3 (see Section 3.8 of [I-D.ietf-lake-edhoc]), EDHOC message_3 has to be transported in the CoAP payload of the EDHOC + OSCORE request.

The rest of this section specifies how to transport the data in the EDHOC + OSCORE request and their processing order. In particular, the use of this approach is explicitly signalled by including an EDHOC Option (see Section 3.1) in the EDHOC + OSCORE request. The processing of the EDHOC + OSCORE request is specified in Section 3.2 for the client side and in Section 3.3 for the server side.

3.1. EDHOC Option

This section defines the EDHOC Option. The option is used in a CoAP request, to signal that the request payload conveys both an EDHOC message_3 and OSCORE-protected data, combined together.

The EDHOC Option has the properties summarized in Figure 3, which extends Table 4 of [RFC7252]. The option is Critical, Safe-to-Forward, and part of the Cache-Key. The option MUST occur at most once and is always empty. If any value is sent, the value is simply ignored. The option is intended only for CoAP requests and is of Class U for OSCORE [RFC8613].

No.	C	U	N	R	Name	Format	Length	Default
TBD21	x				EDHOC	Empty	0	(none)

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

Figure 3: The EDHOC Option.

Note to RFC Editor: Following the registration of the CoAP Option Number 21 as per Section 8.1, please replace "TBD21" with "21" in the figure above. Then, please delete this paragraph.

The presence of this option means that the message payload contains also EDHOC data, that must be extracted and processed as defined in Section 3.3, before the rest of the message can be processed.

Figure 4 shows an example of CoAP message transported over UDP and containing both the EDHOC data and the OSCORE ciphertext, using the newly defined EDHOC option for signalling.

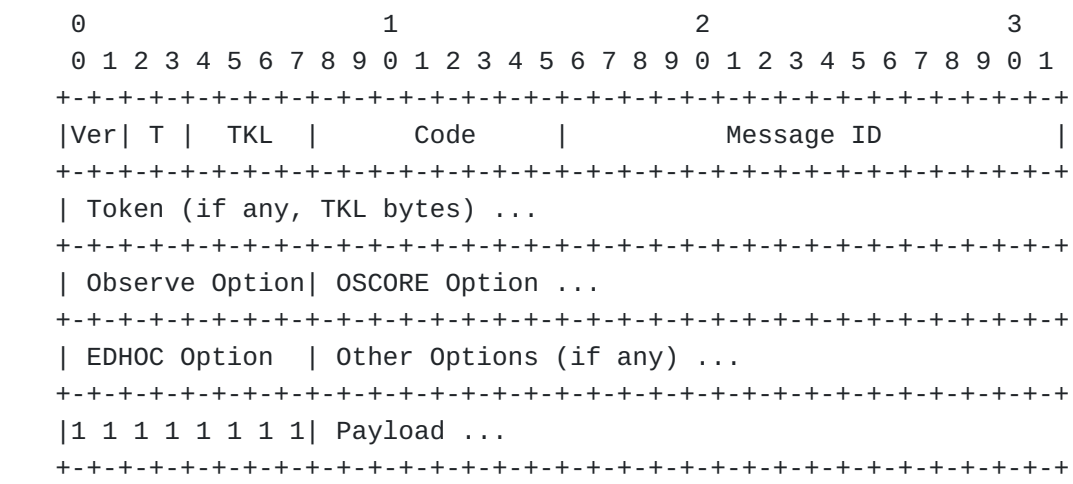

Figure 4: Example of CoAP message transported over UDP, combining EDHOC data and OSCORE data as signalled with the EDHOC Option.

3.2. Client Processing

The client prepares an EDHOC + OSCORE request as follows.

1. Compose EDHOC message_3 as per Section 5.4.2 of [I-D.ietf-lake-edhoc].

2. Establish the new OSCORE Security Context and use it to encrypt the original CoAP request as per Section 8.1 of [RFC8613].

   Note that the OSCORE ciphertext is not computed over EDHOC message_3, which is not protected by OSCORE. That is, the result of this step is the OSCORE Request as in Figure 1.

3. Build COMB_PAYLOAD as the concatenation of EDHOC_MSG_3 and OSCORE_PAYLOAD in this order: COMB_PAYLOAD = EDHOC_MSG_3 | OSCORE_PAYLOAD, where | denotes byte string concatenation and:

   * EDHOC_MSG_3 is the binary encoding of EDHOC message_3 resulting from step 1. As per Section 5.4.1 of [I-D.ietf-lake-edhoc], EDHOC message_3 consists of one CBOR data item CIPHERTEXT_3, which is a CBOR byte string. Therefore, EDHOC_MGS_3 is the binary encoding of CIPHERTEXT_3.

   * OSCORE_PAYLOAD is the OSCORE ciphertext of the OSCORE-protected CoAP request resulting from step 2.

4. Compose the EDHOC + OSCORE request, as the OSCORE-protected CoAP request resulting from step 2, where the payload is replaced with COMB_PAYLOAD built at step 3.

Note that the new payload includes EDHOC message_3, but it does not include the EDHOC connection identifier C_R. As the client is the EDHOC Initiator, C_R is the OSCORE Sender ID of the client, which is already specified as 'kid' in the OSCORE Option of the request from step 2, hence of the EDHOC + OSCORE request.

5. Signal the usage of this approach, by including the new EDHOC Option defined in Section 3.1 into the EDHOC + OSCORE request.

   The application/cid-edhoc+cbor-seq media type does not apply to this message, whose media type is unnamed.

6. Send the EDHOC + OSCORE request to the server.

With the same server, the client SHOULD NOT have multiple simultaneous outstanding interactions (see Section 4.7 of [RFC7252]) such that: they consist of an EDHOC + OSCORE request; and their EDHOC data pertain to the EDHOC session with the same connection identifier C_R.

3.2.1. Supporting Block-wise

If Block-wise [RFC7959] is supported, the client may fragment the first application CoAP request before protecting it as an original message with OSCORE, as defined in Section 4.1.3.4.1 of [RFC8613].

In such a case, the OSCORE processing in step 2 of Section 3.2 is performed on each inner block of the first application CoAP request, and the following also applies.

*The client takes the additional following step between steps 2 and 3 of Section 3.2.

A. If the OSCORE-protected request from step 2 conveys a non-first inner block of the first application CoAP request (i.e., the Block1 Option processed at step 2 had NUM different than 0), then the client skips the following steps and sends the OSCORE-protected request to the server. In particular, the client MUST NOT include the EDHOC Option in the OSCORE-protected request.

*The client takes the additional following step between steps 3 and 4 of Section 3.2.

B. If the size of COMB_PAYLOAD exceeds MAX_UNFRAGMENTED_SIZE (see Section 4.1.3.4.2 of [RFC8613]), the client MUST stop processing the request and MUST abort the Block-wise transfer. Then, the client can continue by switching to the purely sequential workflow shown in Figure 1. That is, the client first sends EDHOC message_3 prepended by the EDHOC Connection Identifier C_R

encoded as per Section 3.3 of [I-D.ietf-lake-edhoc], and then sends the OSCORE-protected CoAP request once the EDHOC execution is completed.

The performance advantage of using the EDHOC + OSCORE request can be lost, when used in combination with Block-wise transfers that rely on specific parameter values and block sizes.

3.3. Server Processing

In order to process a request containing the EDHOC option, i.e., an EDHOC + OSCORE request, the server MUST perform the following steps.

1. Check that the EDHOC + OSCORE request includes the OSCORE option and that the request payload has the format defined at step 3 of Section 3.2 for COMB_PAYLOAD. If this is not the case, the server MUST stop processing the request and MUST reply with a 4.00 (Bad Request) error response.

2. Extract EDHOC message_3 from the payload COMB_PAYLOAD of the EDHOC + OSCORE request, as the first element EDHOC_MSG_3 (see step 3 of Section 3.2).

3. Take the value of 'kid' from the OSCORE option of the EDHOC + OSCORE request (i.e., the OSCORE Sender ID of the client), and use it as the EDHOC connection identifier C_R.

4. Retrieve the correct EDHOC session by using the connection identifier C_R from step 3.

   If the application profile used in the EDHOC session specifies that EDHOC message_4 shall be sent, the server MUST stop the EDHOC processing and consider it failed, as due to a client error.

   Otherwise, perform the EDHOC processing on the EDHOC message_3 extracted at step 2 as per Section 5.4.3 of [I-D.ietf-lake-edhoc], based on the protocol state of the retrieved EDHOC session.

   The application profile used in the EDHOC session is the same one associated with the EDHOC resource where the server received the request conveying EDHOC message_1 that started the session. This is relevant in case the server provides multiple EDHOC resources, which may generally refer to different application profiles.

5. Establish a new OSCORE Security Context associated with the client as per Appendix A.1 of [I-D.ietf-lake-edhoc], using the EDHOC output from step 4.

6. Extract the OSCORE ciphertext from the payload COMB_PAYLOAD of the EDHOC + OSCORE request, as the second element OSCORE_PAYLOAD (see step 3 of Section 3.2).

7. Rebuild the OSCORE-protected CoAP request, as the EDHOC + OSCORE request where the payload is replaced with the OSCORE ciphertext extracted at step 6. Then, remove the EDHOC option.

8. Decrypt and verify the OSCORE-protected CoAP request rebuilt at step 7, as per Section 8.2 of [RFC8613], by using the OSCORE Security Context established at step 5.

   When the decrypted request is checked for any critical CoAP options (as it is during regular CoAP processing), the presence of an EDHOC option MUST be regarded as an unprocessed critical option, unless it is processed by some further mechanism.

9. Deliver the CoAP request resulting from step 8 to the application.

If steps 4 (EDHOC processing) and 8 (OSCORE processing) are both successfully completed, the server MUST reply with an OSCORE-protected response (see Section 5.4.3 of [I-D.ietf-lake-edhoc]). The usage of EDHOC message_4 as defined in Section 5.5 of [I-D.ietf-lake-edhoc] is not applicable to the approach defined in this document.

If step 4 (EDHOC processing) fails, the server discontinues the protocol as per Section 5.4.3 of [I-D.ietf-lake-edhoc] and responds with an EDHOC error message with error code 1, formatted as defined in Section 6.2 of [I-D.ietf-lake-edhoc]. The server MUST NOT establish a new OSCORE Security Context from the present EDHOC session with the client, hence the CoAP response conveying the EDHOC error message is not protected with OSCORE. As per Section 8.5 of [I-D.ietf-lake-edhoc], the server has to make sure that the error message does not reveal sensitive information. The CoAP response conveying the EDHOC error message MUST have Content-Format set to application/edhoc+cbor-seq defined in Section 9.9 of [I-D.ietf-lake-edhoc].

If step 4 (EDHOC processing) is successfully completed but step 8 (OSCORE processing) fails, the same OSCORE error handling as defined in Section 8.2 of [RFC8613] applies.

3.3.1. Supporting Block-wise

If Block-wise [RFC7959] is supported, the server takes the additional following step before any other in Section 3.3.

A. If Block-wise is present in the request, then process the Outer Block options according to [RFC7959], until all blocks of the request have been received (see Section 4.1.3.4 of [RFC8613]).

3.4. Example of EDHOC + OSCORE Request

Figure 5 shows an example of EDHOC + OSCORE Request transported over UDP. In particular, the example assumes that:

- The used OSCORE Partial IV is 0, consistently with the first request protected with the new OSCORE Security Context.

- The OSCORE Sender ID of the client is 0x01.

  As per Section 3.3.3 of [I-D.ietf-lake-edhoc], this straightforwardly corresponds to the EDHOC connection identifier C_R 0x01.

  As per Section 3.3.2 of [I-D.ietf-lake-edhoc], when using the purely-sequential flow shown in Figure 1, the same C_R with value 0x01 would be encoded on the wire as the CBOR integer 1 (0x01 in CBOR encoding), and prepended to EDHOC message_3 in the payload of the second EDHOC request.

- The EDHOC option is registered with CoAP option number 21.

Note to RFC Editor: Please delete the last bullet point in the previous list, since, at the time of publication, the CoAP option number will be in fact registered.

- OSCORE option value: 0x090001 (3 bytes)

- EDHOC option value: - (0 bytes)

- EDHOC message_3: 0x52d5535f3147e85f1cfacd9e78abf9e0a81bbf (19 bytes)

- OSCORE ciphertext: 0x612f1092f1776f1c1668b3825e (13 bytes)

From there:

- Protected CoAP request (OSCORE message):

```
0x44025d1f               ; CoAP 4-byte header
  00003974               ; Token
  39 6c6f63616c686f7374  ; Uri-Host Option: "localhost"
  63 090001              ; OSCORE Option
  c0                     ; EDHOC Option
  ff 52d5535f3147e85f1cfacd9e78abf9e0a81bbf
     612f1092f1776f1c1668b3825e
(56 bytes)
```

Figure 5: Example of CoAP message transported over UDP, combining EDHOC data and OSCORE data as signalled with the EDHOC Option.

4. Use of EDHOC Connection Identifiers with OSCORE

Section 3.3.3 of [I-D.ietf-lake-edhoc] defines the straightforward mapping from an EDHOC connection identifier to an OSCORE Sender/Recipient ID. That is, an EDHOC identifier and the corresponding OSCORE Sender/Recipient ID are both byte strings with the same value.

Therefore, the conversion from an OSCORE Sender/Recipient ID to an EDHOC identifier is equally straightforward. In particular, at step 3 of Section 3.3, the value of 'kid' in the OSCORE Option of the EDHOC + OSCORE request is both the server's Recipient ID (i.e., the client's Sender ID) as well as the EDHOC Connection Identifier C_R of the server.

4.1. Additional Processing of EDHOC Messages

When using EDHOC to establish an OSCORE Security Context, the client and server MUST perform the following additional steps during an EDHOC execution, thus extending Section 5 of [I-D.ietf-lake-edhoc].

4.1.1. Initiator Processing of Message 1

The Initiator selects an EDHOC Connection Identifier C_I as follows.

The Initiator MUST choose a C_I that is neither used in any current EDHOC session as this peer's EDHOC Connection Identifier, nor the Recipient ID in a current OSCORE Security Context where the ID Context is not present.

The chosen C_I SHOULD NOT be the Recipient ID of any current OSCORE Security Context.

4.1.2. Responder Processing of Message 2

The Responder selects an EDHOC Connection Identifier C_R as follows.

The Responder MUST choose a C_R that is neither used in any current EDHOC session as this peer's EDHOC Connection Identifier, nor is equal to the EDHOC Connection Identifier C_I specified in the EDHOC message_1 of the present EDHOC session (i.e., after its decoding as per Section 3.3 of [I-D.ietf-lake-edhoc]), nor is the Recipient ID in a current OSCORE Security Context where the ID Context is not present.

The chosen C_R SHOULD NOT be the Recipient ID of any current OSCORE Security Context.

4.1.3. Initiator Processing of Message 2

If the following condition holds, the Initiator MUST discontinue the protocol and reply with an EDHOC error message with error code 1, formatted as defined in Section 6.2 of [I-D.ietf-lake-edhoc].

* The EDHOC Connection Identifier C_I is equal to the EDHOC Connection Identifier C_R specified in EDHOC message_2 (i.e., after its decoding as per Section 3.3 of [I-D.ietf-lake-edhoc]).

5. Extension and Consistency of Application Profiles

The application profile referred by the client and server can include the information elements introduced below, in accordance with the specified consistency rules.

If the server supports the EDHOC + OSCORE request within an EDHOC execution started at a certain EDHOC resource, then the application profile associated with that resource:

* MUST NOT specify that EDHOC message_4 shall be sent.

* SHOULD explicitly specify support for the EDHOC + OSCORE request.

6. Web Linking

Section 9.10 of [I-D.ietf-lake-edhoc] registers the resource type "core.edhoc", which can be used as target attribute in a web link [RFC8288] to an EDHOC resource, e.g., using a link-format document [RFC6690]. This enables clients to discover the presence of EDHOC resources at a server, possibly using the resource type as filter criterion.

At the same time, the application profile associated with an EDHOC resource provides a number of information describing how the EDHOC protocol can be used through that resource. While a client may become aware of the application profile through several means, it would be convenient to obtain its information elements upon discovering the EDHOC resources at the server. This might aim at discovering especially the EDHOC resources whose associated application profile denotes a way of using EDHOC which is most suitable to the client, e.g., with EDHOC cipher suites or authentication methods that the client also supports or prefers.

That is, it would be convenient that a client discovering an EDHOC resource contextually obtains relevant pieces of information from the application profile associated with that resource. The resource discovery can occur by means of a direct interaction with the server, or instead by means of the CoRE Resource Directory [RFC9176], where the server may have registered the links to its resources.

In order to enable the above, this section defines a number of parameters, each of which can be optionally specified as a target attribute with the same name in the link to the respective EDHOC resource, or as filter criteria in a discovery request from the client. When specifying these parameters in a link to an EDHOC resource, the target attribute rt="core.edhoc" MUST be included, and the same consistency rules defined in Section 5 for the corresponding information elements of an application profile MUST be followed.

The following parameters are defined.

* 'ed-i', specifying, if present, that the server supports the EDHOC Initiator role, hence the reverse message flow of EDHOC. A value MUST NOT be given to this parameter and any present value MUST be ignored by parsers.

* 'ed-r', specifying, if present, that the server supports the EDHOC Responder role, hence the forward message flow of EDHOC. A value MUST NOT be given to this parameter and any present value MUST be ignored by parsers.

* 'ed-method', specifying an authentication method supported by the server. This parameter MUST specify a single value, which is taken from the 'Value' column of the "EDHOC Method Type" registry defined in Section 9.3 of [I-D.ietf-lake-edhoc]. This parameter MAY occur multiple times, with each occurrence specifying an authentication method.

* 'ed-csuite', specifying an EDHOC cipher suite supported by the server. This parameter MUST specify a single value, which is taken from the 'Value' column of the "EDHOC Cipher Suites" registry defined in Section 9.2 of [I-D.ietf-lake-edhoc]. This parameter MAY occur multiple times, with each occurrence specifying a cipher suite.

* 'ed-cred-t', specifying a type of authentication credential supported by the server. This parameter MUST specify a single value, which is taken from the 'Value' column of the "EDHOC Authentication Credential Types" Registry defined in Section 8.3 of this document. This parameter MAY occur multiple times, with each occurrence specifying a type of authentication credential.

* 'ed-idcred-t', specifying a type of identifier supported by the server for identifying authentication credentials. This parameter MUST specify a single value, which is taken from the 'Label' column of the "COSE Header Parameters" registry [COSE.Header.Parameters]. This parameter MAY occur multiple times, with each occurrence specifying a type of identifier for authentication credentials.

  Note that the values in the 'Label' column of the "COSE Header Parameters" registry are strongly typed. On the contrary, Link Format is weakly typed and thus does not distinguish between, for instance, the string value "-10" and the integer value -10. Thus, if responses in Link Format are returned, string values which look like an integer are not supported. Therefore, such values MUST NOT be used in the 'ed-idcred-t' parameter.

* 'ed-ead', specifying the support of the server for an External Authorization Data (EAD) item (see Section 3.8 of [I-D.ietf-lake-edhoc]). This parameter MUST specify a single value, which is taken from the 'Label' column of the "EDHOC External Authorization Data" registry defined in Section 9.5 of [I-D.ietf-lake-edhoc]. This parameter MAY occur multiple times, with each occurrence specifying the ead_label of an EAD item that the server supports.

* 'ed-comb-req', specifying, if present, that the server supports the EDHOC + OSCORE request defined in Section 3. A value MUST NOT

be given to this parameter and any present value MUST be ignored by parsers.

The example in Figure 6 shows how a client discovers one EDHOC resource at a server, obtaining information elements from the respective application profile. The Link Format notation from Section 5 of [RFC6690] is used.

```
REQ: GET /.well-known/core

RES: 2.05 Content
    </sensors/temp>;osc,
    </sensors/light>;if=sensor,
    </.well-known/edhoc>;rt=core.edhoc;ed-csuite=0;ed-csuite=2;
        ed-method=0;ed-cred-t=1;ed-cred-t=3;ed-idcred-t=4;
        ed-i;ed-r;ed-comb-req
```

Figure 6: The Web Link.

7. Security Considerations

The same security considerations from OSCORE [RFC8613] and EDHOC [I-D.ietf-lake-edhoc] hold for this document. In addition, the following considerations also apply.

Section 3.2 specifies that a client SHOULD NOT have multiple outstanding EDHOC + OSCORE requests pertaining to the same EDHOC session. Even if a client did not fulfill this requirement, it would not have any impact in terms of security. That is, the server would still not process different instances of the same EDHOC message_3 more than once in the same EDHOC session (see Section 5.1 of [I-D.ietf-lake-edhoc]), and would still enforce replay protection of the OSCORE-protected request (see Sections 7.4 and 8.2 of [RFC8613]).

When using the optimized workflow in Figure 2, a minimum of 128-bit security against online brute force attacks is achieved after the client receives and successfully verifies the first OSCORE-protected response (see Section 8.1 of [I-D.ietf-lake-edhoc]). As an example, if EDHOC is used with method 3 (see Section 3.2 of [I-D.ietf-lake-edhoc]) and cipher suite 2 (see Section 3.6 of [I-D.ietf-lake-edhoc]), then the following holds.

* The Initiator is authenticated with 128-bit security against online attacks. This is the sum of the 64-bit MACs in EDHOC message_3 and of the MAC in the AEAD of the first OSCORE-protected CoAP request, as rebuilt at step 7 of Section 3.3.

*The Responder is authenticated with 128-bit security against online attacks. This is the sum of the 64-bit MACs in EDHOC message_2 and of the MAC in the AEAD of the first OSCORE-protected CoAP response.

With reference to the purely sequential workflow in Figure 1, the OSCORE request might have to undergo access control checks at the server, before being actually executed for accesing the target protected resource. The same MUST hold when the optimized workflow in Figure 2 is used, i.e., when using the EDHOC + OSCORE request.

That is, the rebuilt OSCORE-protected application request from step 7 in Section 3.3 MUST undergo the same access control checks that would be performed on a traditional OSCORE-protected application request sent individually as shown in Figure 1.

To this end, validated information to perform access control checks (e.g., an access token issued by a trusted party) has to be available at the server latest before starting to process the rebuilt OSCORE-protected application request. Such information may have been provided to the server separately before starting the EDHOC execution altogether, or instead as External Authorization Data during the EDHOC execution (see Section 3.8 of [I-D.ietf-lake-edhoc]).

Thus, a successful completion of the EDHOC protocol and the following derivation of the OSCORE Security Context at the server do not play a role in determining whether the rebuilt OSCORE-protected request is authorized to access the target protected resource at the server.

8. IANA Considerations

This document has the following actions for IANA.

Note to RFC Editor: Please replace all occurrences of "[RFC-XXXX]" with the RFC number of this specification and delete this paragraph.

8.1. CoAP Option Numbers Registry

IANA is asked to enter the following option number to the "CoAP Option Numbers" registry within the "CoRE Parameters" registry group.

Number	Name	Reference
TBD21	EDHOC	[RFC-XXXX]

Note to RFC Editor: Following the registration of the CoAP Option Number 21, please replace "TBD21" with "21" in the table above. Then, please delete this paragraph and all the following text within the present Section 8.1.

[

The CoAP option number 21 is consistent with the properties of the EDHOC Option defined in Section 3.1, and it allows the EDHOC Option to always result in an overall size of 1 byte. This is because:

- The EDHOC option is always empty, i.e., with zero-length value; and

- Since the OSCORE Option with option number 9 is always present in the EDHOC + OSCORE request, the EDHOC Option is encoded with a delta equal to at most 12.

Therefore, this document suggests 21 (TBD21) as option number to be assigned to the new EDHOC Option. Although the currently unassigned option number 13 would also work well for the same reasons in the use case in question, different use cases or protocols may make a better use of the option number 13. Hence the preference for the option number 21, and why it is *not* necessary to register additional option numbers than 21.

]

8.2. Target Attributes Registry

IANA is asked to register the following entries in the "Target Attributes" registry within the "CoRE Parameters" registry group, as per [I-D.ietf-core-target-attr].

```
Attribute Name: ed-i
Brief Description: Hint: support for the EDHOC Initiator role
Change Controller: IESG
Reference: [RFC-XXXX]

Attribute Name: ed-r
Brief Description: Hint: support for the EDHOC Responder role
Change Controller: IESG
Reference: [RFC-XXXX]

Attribute Name: ed-method
Brief Description: A supported authentication method for EDHOC
Change Controller: IESG
Reference: [RFC-XXXX]

Attribute Name: ed-csuite
Brief Description: A supported cipher suite for EDHOC
Change Controller: IESG
Reference: [RFC-XXXX]

Attribute Name: ed-cred-t
Brief Description: A supported type of
                   authentication credential for EDHOC
Change Controller: IESG
Reference: [RFC-XXXX]

Attribute Name: ed-idcred-t
Brief Description: A supported type of
                   authentication credential identifier for EDHOC
Change Controller: IESG
Reference: [RFC-XXXX]

Attribute Name: ed-ead
Brief Description: A supported External Authorization Data (EAD)
                   item for EDHOC
Change Controller: IESG
Reference: [RFC-XXXX]

Attribute Name: ed-comb-req
Brief Description: Hint: support for the EDHOC+OSCORE request
Change Controller: IESG
Reference: [RFC-XXXX]
```

8.3. EDHOC Authentication Credential Types Registry

IANA is requested to create a new "EDHOC Authentication Credential Types" registry within the "Ephemeral Diffie-Hellman Over COSE (EDHOC)" registry group defined in [I-D.ietf-lake-edhoc].

The registry uses the "Expert Review" registration procedure [RFC8126]. Expert Review guidelines are provided in Section 8.4.

The columns of this registry are:

- *Value: This field contains the value used to identify the type of authentication credential. These values MUST be unique. The value can be an unsigned integer or a negative integer. Different ranges of values use different registration policies [RFC8126]. Integer values from -24 to 23 are designated as "Standards Action With Expert Review". Integer values from -65536 to -25 and from 24 to 65535 are designated as "Specification Required". Integer values smaller than -65536 and greater than 65535 are marked as "Private Use".

- *Description: This field contains a short description of the type of authentication credential.

- *Reference: This field contains a pointer to the public specification for the type of authentication credential.

Initial entries in this registry are as listed in Figure 7.

Value	Description	Reference
0	CBOR Web Token (CWT) containing a COSE_Key in a 'cnf' claim	[RFC8392]
1	CWT Claims Set (CCS) containing a COSE_Key in a 'cnf' claim	[RFC8392]
2	X.509 certificate	[RFC5280]
3	C509 certificate	[I-D.ietf-cose-cbor-encoded-cert]

Figure 7: Initial Entries in the "EDHOC Authentication Credential Types" Registry

8.4. Expert Review Instructions

The IANA registry established in this document is defined as "Expert Review". This section gives some general guidelines for what the experts should be looking for, but they are being designated as experts for a reason so they should be given substantial latitude.

Expert reviewers should take into consideration the following points:

* Clarity and correctness of registrations. Experts are expected to check the clarity of purpose and use of the requested entries. Experts need to make sure that registered identifiers indicate a type of authentication credential whose format and encoding is clearly defined in the corresponding specification. Identifiers of types of authentication credentials that do not meet these objective of clarity and completeness must not be registered.

* Point squatting should be discouraged. Reviewers are encouraged to get sufficient information for registration requests to ensure that the usage is not going to duplicate one that is already registered and that the point is likely to be used in deployments. The zones tagged as "Private Use" are intended for testing purposes and closed environments. Code points in other ranges should not be assigned for testing.

* Specifications are required for the "Standards Action With Expert Review" range of point assignment. Specifications should exist for "Specification Required" ranges, but early assignment before a specification is available is considered to be permissible. When specifications are not provided, the description provided needs to have sufficient information to identify what the point is being used for.

* Experts should take into account the expected usage of fields when approving point assignment. The fact that there is a range for Standards Track documents does not mean that a Standards Track document cannot have points assigned outside of that range. The length of the encoded value should be weighed against how many code points of that length are left, the size of device it will be used on, and the number of code points left that encode to that size.

9. References

9.1. Normative References

[COSE.Header.Parameters] IANA, "COSE Header Parameters", <https://www.iana.org/assignments/cose/cose.xhtml#header-parameters>.

[I-D.ietf-core-target-attr] Bormann, C., "CoRE Target Attributes Registry", Work in Progress, Internet-Draft, draft-ietf-core-target-attr-04, 5 March 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-core-target-attr-04>.

[I-D.ietf-lake-edhoc] Selander, G., Mattsson, J. P., and F. Palombini, "Ephemeral Diffie-Hellman Over COSE (EDHOC)", Work in Progress, Internet-Draft, draft-ietf-lake-edhoc-19, 3 February 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-lake-edhoc-19>.

[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>.

[RFC6690] Shelby, Z., "Constrained RESTful Environments (CoRE) Link Format", RFC 6690, DOI 10.17487/RFC6690, August 2012, <https://www.rfc-editor.org/info/rfc6690>.

[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>.

[RFC7959] Bormann, C. and Z. Shelby, Ed., "Block-Wise Transfers in the Constrained Application Protocol (CoAP)", RFC 7959, DOI 10.17487/RFC7959, August 2016, <https://www.rfc-editor.org/info/rfc7959>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[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>.

[RFC8288] Nottingham, M., "Web Linking", RFC 8288, DOI 10.17487/RFC8288, October 2017, <https://www.rfc-editor.org/info/rfc8288>.

[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>.

[RFC8949] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", STD 94, RFC 8949, DOI 10.17487/

RFC8949, December 2020, <https://www.rfc-editor.org/info/rfc8949>.

[RFC9176] Amsüss, C., Ed., Shelby, Z., Koster, M., Bormann, C., and P. van der Stok, "Constrained RESTful Environments (CoRE) Resource Directory", RFC 9176, DOI 10.17487/RFC9176, April 2022, <https://www.rfc-editor.org/info/rfc9176>.

9.2. Informative References

[I-D.ietf-cose-cbor-encoded-cert]
Mattsson, J. P., Selander, G., Raza, S., Höglund, J., and M. Furuhed, "CBOR Encoded X.509 Certificates (C509 Certificates)", Work in Progress, Internet-Draft, draft-ietf-cose-cbor-encoded-cert-05, 10 January 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-cose-cbor-encoded-cert-05>.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <https://www.rfc-editor.org/info/rfc5280>.

[RFC8392] Jones, M., Wahlstroem, E., Erdtman, S., and H. Tschofenig, "CBOR Web Token (CWT)", RFC 8392, DOI 10.17487/RFC8392, May 2018, <https://www.rfc-editor.org/info/rfc8392>.

Appendix A. Document Updates

RFC Editor: Please remove this section.

A.1. Version -06 to -07

* Changed document title.

* The client creates the OSCORE Security Context after creating EDHOC message_3.

* Revised selection of EDHOC connection identifiers.

* Use of "forward message flow" and "reverse message flow".

* The payload of the combined request is not a CBOR sequence anymore.

* EDHOC error messages from the server are not protected with OSCORE.

*More future-proof error handling on the server side.

*Target attribute names prefixed by "ed-".

*Defined new target attributes "ed-i" and "ed-r".

*Defined single target attribute "ed-ead" signaling supported EAD items.

*Security consideration on the minimally achieved 128-bit security.

*Defined and used the "EDHOC Authentication Credential Types" Registry.

*High-level sentence replacing the appendix on Block-wise performance.

*Revised examples.

*Editorial improvements.

A.2. Version -05 to -06

*Extended figure on EDHOC sequential workflow.

*Revised naming of target attributes.

*Clarified semantics of target attributes 'eadx'.

*Registration of target attributes.

A.3. Version -04 to -05

*Clarifications on Web Linking parameters.

*Added security considerations.

*Revised IANA considerations to focus on the CoAP option number 21.

*Guidelines on using Block-wise moved to an appendix.

*Editorial improvements.

A.4. Version -03 to -04

*Renamed "applicability statement" to "application profile".

*Use the latest Content-Formats.

*Use of SHOULD NOT for multiple simultaneous outstanding interactions.

*No more special conversion from OSCORE ID to EDHOC ID.

*Considerations on using Block-wise.

*Wed Linking signaling of multiple supported EAD labels.

*Added security considerations.

*Editorial improvements.

A.5. Version -02 to -03

*Clarifications on transporting EDHOC message_3 in the CoAP payload.

*At most one simultaneous outstanding interaction as an EDHOC + OSCORE request with the same server for the same session with connection identifier C_R.

*The EDHOC option is removed from the EDHOC + OSCORE request after processing the EDHOC data.

*Added explicit constraints when selecting a Recipient ID as C_X.

*Added processing steps for when Block-wise is used.

*Improved error handling on the server.

*Improved section on Web Linking.

*Updated figures; editorial improvements.

A.6. Version -01 to -02

*New title, abstract and introduction.

*Restructured table of content.

*Alignment with latest format of EDHOC messages.

*Guideline on ID conversions based on application profile.

*Clarifications, extension and consistency on application profile.

*Section on web-linking.

*RFC8126 terminology in IANA considerations.

*Revised Appendix "Checking CBOR Encoding of Numeric Values".

A.7. Version -00 to -01

*Improved background overview of EDHOC.

*Added explicit rules for converting OSCORE Sender/Recipient IDs to EDHOC connection identifiers following the removal of bstr_identifier from EDHOC.

*Revised section organization.

*Recommended number for EDHOC option changed to 21.

*Editorial improvements.

Acknowledgments

The authors sincerely thank Christian Amsüss, Esko Dijk, Klaus Hartke, John Preuß Mattsson, David Navarro, Jim Schaad and Mališa Vučinić for their feedback and comments.

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

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@eriptic.com

Goeran Selander
Ericsson

Email: goran.selander@ericsson.com