Workgroup: LAKE Working Group
Internet-Draft: draft-selander-lake-authz-03
Published: 7 July 2023
Intended Status: Standards Track
Expires: 8 January 2024
Authors: G. Selander J. Preuß Mattsson M. Vučinić
Ericsson AB Ericsson AB INRIA
M. Richardson A. Schellenbaum
Sandelman Software Works ZHAW

Lightweight Authorization using Ephemeral Diffie-Hellman Over COSE

Abstract

This document describes a procedure for authorizing enrollment of new devices using the lightweight authenticated key exchange protocol Ephemeral Diffie-Hellman Over COSE (EDHOC). The procedure is applicable to zero-touch onboarding of new devices to a constrained network leveraging trust anchors installed at manufacture time.

About This Document

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

The latest revision of this draft can be found at https://ericssonresearch.github.io/ace-ake-authz/draft-selander-lake-authz.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-selander-lake-authz/.

Discussion of this document takes place on the Lightweight Authenticated Key Exchange Working Group mailing list (mailto:lake@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/lake/. Subscribe at https://www.ietf.org/mailman/listinfo/lake/.

Source for this draft and an issue tracker can be found at https://github.com/EricssonResearch/ace-ake-authz.

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 8 January 2024.

Copyright Notice

Table of Contents

1. Introduction

For constrained IoT deployments [RFC7228] the overhead and processing contributed by security protocols may be significant which motivates the specification of lightweight protocols that are optimizing, in particular, message overhead (see [I-D.ietf-lake-reqs]). This document describes a procedure for augmenting the lightweight authenticated Diffie-Hellman key exchange EDHOC [I-D.ietf-lake-edhoc] with third party-assisted authorization.

The procedure involves a device, a domain authenticator, and an enrollment server. The device and domain authenticator perform mutual authentication and authorization, assisted by the enrollment server which provides relevant authorization information to the device (a "voucher") and to the authenticator. The high-level model is similiar to BRSKI [RFC8995].

In this document we consider the target interaction for which authorization is needed to be "enrollment", for example joining a network for the first time (e.g., [RFC9031]), but it can be applied to authorize other target interactions.

The enrollment server may represent the manufacturer of the device, or some other party with information about the device from which a trust anchor has been pre-provisioned into the device. The (domain) authenticator may represent the service provider or some other party controlling access to the network in which the device is enrolling.

The protocol assumes that authentication between device and authenticator is performed with EDHOC [I-D.ietf-lake-edhoc], and defines the integration of a lightweight authorization procedure using the External Authorization Data (EAD) fields defined in EDHOC.

The protocol enables a low message count by performing authorization and enrollment in parallel with authentication, instead of in sequence which is common for network access. It further reuses protocol elements from EDHOC leading to reduced message sizes on constrained links.

This protocol is applicable to a wide variety of settings, and can be mapped to different authorization architectures.

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.

Readers are expected to have an understanding of CBOR [RFC8949] and EDHOC [I-D.ietf-lake-edhoc]. Appendix C.1 of [I-D.ietf-lake-edhoc] contains some basic info about CBOR.

2. Problem Description

The (potentially constrained) device (U) wants to enroll into a domain over a constrained link. The device authenticates and enforces authorization of the (non-constrained) domain authenticator (V) with the help of a voucher conveying authorization information. The domain authenticator, in turn, authenticates the device and authorizes its enrollment into the domain.

The procedure is assisted by a (non-constrained) enrollment server (W) located in a non-constrained network behind the domain authenticator, e.g. on the Internet, providing information to the device (the voucher) and to the domain authenticator as part of the protocol.

The objective of this document is to specify such a protocol which is lightweight over the constrained link by reusing elements of EDHOC [I-D.ietf-lake-edhoc] and by shifting message overhead to the non-constrained side of the network. See illustration in Figure 1.

Note the cardinality of the involved parties. It is expected that the authenticator needs to handle a large unspecified number of devices, but for a given device type or manufacturer it is expected that one or a few nodes host enrollment servers.

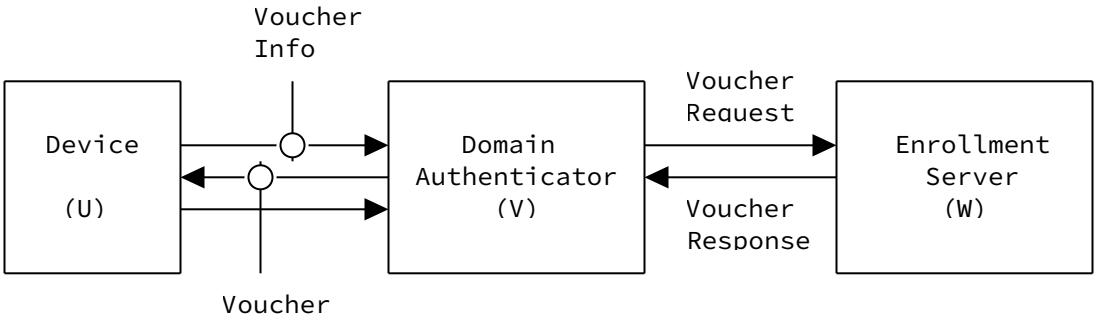

Figure 1: Overview of message flow. EDHOC is used on the constrained link between U and V. Voucher Info and Voucher are sent in EDHOC External Authorization Data (EAD). The link between V and W is not constrained.

3. Assumptions

The protocol is based on the following pre-existing relations between the device (U), the domain authenticator (V) and the enrollment server (W), see Figure 2.

- *U and W have an explicit relation: U is configured with a public key of W, see Section 3.1.
- *V and W have an implicit relation, e.g., based on web PKI with trusted CA certificates, see Section 3.2.
- *U and V need not have any previous relation, this protocol establishes a relation between U and V.

Each of the three parties have protected communication with the other two during the protocol.

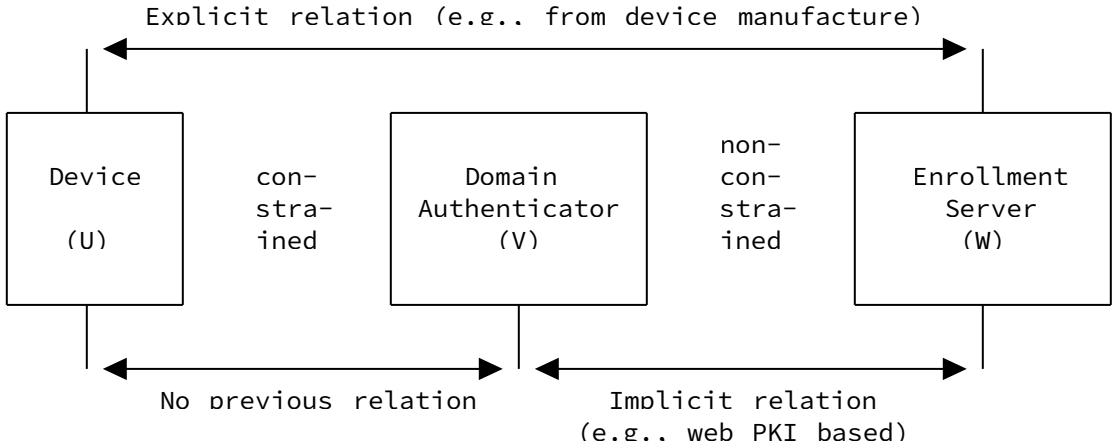

Figure 2: Overview of pre-existing relations.

3.1. Device (U)

To authenticate to V, the device (U) runs EDHOC in the role of Initiator with authentication credential CRED_U, for example, an X.509 certificate or a CBOR Web Token (CWT, [RFC8392]). CRED_U may, for example, be carried in ID_CRED_I of EDHOC message_3 or be provisioned to V over a non-constrained network, see bottom of Figure 3.

U also needs to identify itself to W, this device identifier is denoted by ID_U. The purpose of ID_U is for W to be able to determine if the device with this identifier is authorized to enroll with V. ID_U may be a reference to CRED_U, like ID_CRED_I in EDHOC (see Section 3.5.2 of [I-D.ietf-lake-edhoc]), or a device identifier from a different name space, such as EUI-64 identifiers.

U is also provisioned with information about W:

* A static public DH key of W (G_W) used to establish secure communication with the enrollment server (see Section 4.4).

* Location information about the enrollment server (LOC_W) that can be used by V to reach W. This is typically a URI but may alternatively be only the domain name.

3.2. Domain Authenticator (V)

To authenticate to U, the domain authenticator (V) runs EDHOC in the role of Responder with an authentication credential CRED_V, which is a CWT Claims Set [RFC8392] containing a public key of V, see Section 4.5.2.1. This proves to U the possession of the private key corresponding to the public key of CRED_V. CRED_V typically needs to be transported to U in EDHOC (using ID_CRED_R = CRED_V, see Section 3.5.2 of [I-D.ietf-lake-edhoc]) since there is no previous relation between U and V.

V and W need to establish a secure (confidentiality and integrity protected) connection for the Voucher Request/Response protocol. Furthermore, W needs access the same credential CRED_V as V used with U, and V needs to prove to W the possession of the private key corresponding to the public key of CRED_V. It is RECOMMENDED that V authenticates to W using the same credential CRED_V as with U.

* V and W may protect the Voucher Request/Response protocol using TLS 1.3 with client authentication [RFC8446] if CRED_V is an X.509 certificate of a signature public key. However, note that CRED_V may not be a valid credential to use with TLS 1.3, e.g., when U and V run EDHOC with method 1 or 3, where the public key of CRED_V is a static Diffie-Hellman key.

* V may run EDHOC with W using ID_CRED_I = CRED_V. In this case the secure connection between V and W may be based on OSCORE [RFC8613].

Note that both TLS 1.3 and EDHOC may be run between V and W during this setup procedure. For example, W may authenticate to V using TLS 1.3 with server certificates signed by a CA trusted by V, and then V may run EDHOC using CRED_V over the secure TLS connection to W, see Figure 3.

Note also that the secure connection between V and W may be long lived and reused for multiple voucher requests/responses.

Other details of proof-of-possession related to CRED_V and transport of CRED_V are out of scope of this document.

3.3. Enrollment Server (W)

The enrollment server (W) is assumed to have the private DH key corresponding to G_W, which is used to establish secure communication with the device (see Section 4.4). W provides to U the authorization decision for enrollment with V in the form of a voucher (see Section 4.4.2). Authorization policies are out of scope for this document.

Authentication credentials and communication security with V is described in Section 3.2. To calculate the voucher, W needs access to message_1 and CRED_V as used in the EDHOC session between U and V, see Section 4.4.2.

- *W MUST verify that CRED_V is bound to the secure connection between W and V

- *W MUST verify that V is in possession of the private key corresponding to the public key of CRED_V

W needs to be available during the execution of the protocol between U and V.

4. The Protocol

4.1. Overview

The protocol consist of three security sessions going on in parallel:

1. The EDHOC session between device (U) and (domain) authenticator (V)

2. Voucher Request/Response between authenticator (V) and enrollment server (W)

3. An exchange of voucher-related information, including the voucher itself, between device (U) and enrollment server (W), mediated by the authenticator (V).

Figure 3 provides an overview of the message flow detailed in this section. An outline of EDHOC is given in Section 3 of [I-D.ietf-lake-edhoc].

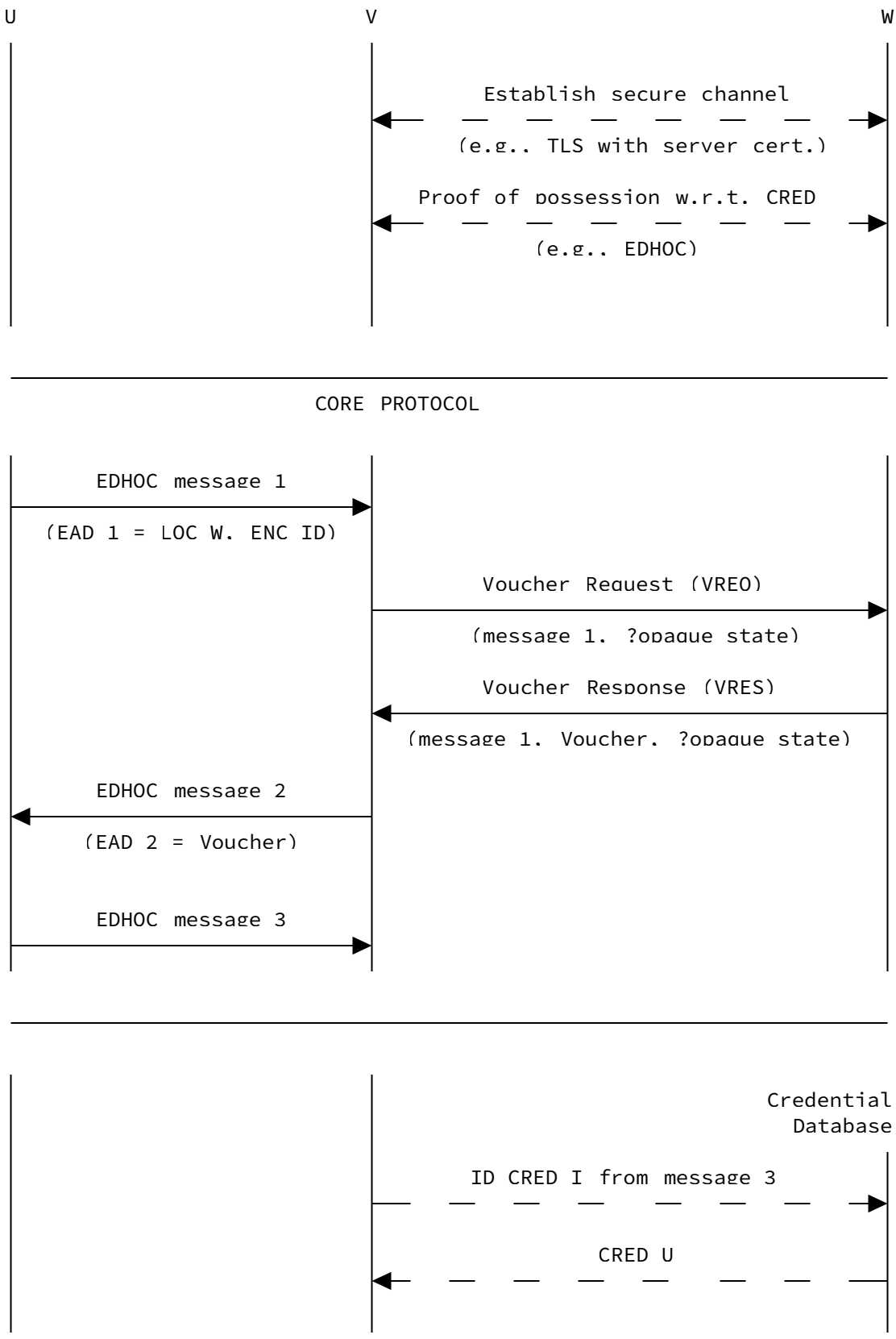
U
V
W
Establish secure channel
(e.g., TLS with server cert.)
Proof of possession w.r.t. CRED
(e.g., EDHOC)
CORE PROTOCOL
EDHOC message 1
(EAD 1 = LOC W, ENC ID)
Voucher Request (VREQ)
(message 1, ?opaque state)
Voucher Response (VRES)
(message 1, Voucher, ?opaque state)
EDHOC message 2
(EAD 2 = Voucher)
EDHOC message 3
Credential
Database
ID CRED I from message 3
CRED U

Figure 3: Overview of the protocol: W-assisted authorization of U and V to each other: EDHOC between U and V, and Voucher Request/Response between V and W. Before the protocol, V and W are assumed to have established a secure channel and performed proof-of-possession of relevant keys. Credential lookup of CRED_U may involve W or other credential database.

4.2. Reuse of EDHOC

The protocol illustrated in Figure 3 reuses several components of EDHOC:

* G_X, the ephemeral public Diffie-Hellman key of U, is also used in the protocol between U and W.

* SUITES_I includes the cipher suite for EDHOC selected by U, and also defines the algorithms used between U and W (see Section 3.6 of [I-D.ietf-lake-edhoc]):

  - EDHOC AEAD algorithm: used to encrypt ID_U

  - EDHOC hash algorithm: used for key derivation and to calculate the voucher

  - EDHOC MAC length in bytes: length of the voucher

  - EDHOC key exchange algorithm: used to calculate the shared secret between U and W

* EAD_1, EAD_2 are the External Authorization Data message fields of message_1 and message_2, respectively, see Section 3.8 of [I-D.ietf-lake-edhoc]. This document specifies the EAD items with ead_label = TBD1, see Section 7.1).

* ID_CRED_I and ID_CRED_R are used to identify the authentication credentials CRED_U and CRED_V, respectively. As shown at the bottom of Figure 3, V may use W to obtain CRED_U. CRED_V is transported in ID_CRED_R in message_2, see Section 4.5.2.1.

The protocol also reuses the EDHOC-Extract and EDHOC-Expand key derivation from EDHOC (see Section 4 of [I-D.ietf-lake-edhoc]).

* The intermediate pseudo-random key PRK is derived using EDHOC-Extract():

  - PRK = EDHOC-Extract(salt, IKM)

    owhere salt = 0x (the zero-length byte string)

oIKM is computed as an ECDH cofactor Diffie-Hellman shared secret from the public key of W, G_W, and the private key corresponding to G_X (or v.v.), see Section 5.7.1.2 of [NIST-800-56A].

The output keying material OKM is derived from PRK using EDHOC-Expand(), which is defined in terms of the EDHOC hash algorithm of the selected cipher suite, see Section 4.2 of [I-D.ietf-lake-edhoc]:

*OKM = EDHOC-Expand(PRK, info, length)

where

```
info = (
   info_label : int,
   context : bstr,
   length : uint,
)
```

4.3. Stateless Operation of V

V may act statelessly with respect to U: the state of the EDHOC session started by U may be dropped at V until authorization from W is received. Once V has received EDHOC message_1 from U and extracted LOC_W from EAD_1, message_1 is forwarded unmodified to W in the form of a Voucher Request. V encapsulates the internal state that it needs to later respond to U, and sends that to W together with EDHOC message_1. This state typically contains U's IP address and port number, together with any other implementation-specific parameter needed by V to respond to U. At this point, V can drop the EDHOC session that was initiated by U.

V MUST encrypt and integrity protect the encapsulated state using a uniformly-distributed (pseudo-)random key, known only to itself. How V serializes and encrypts its internal state is out of scope of this specification. For example, V may use the existing CBOR and COSE libraries.

Editor's note: Consider to include an example of serialized internal state.

W sends to V the voucher together with echoed message_1, as received from U, and V's internal state. This allows V to act as a simple message relay until it has obtained the authorization from W to enroll U. The reception of a successful Voucher Response at V from W implies the authorization for V to enroll U. At this point, V can initialize a new EDHOC session with U, based on the message and the state retrieved from the Voucher Response from W.

4.4. Device <-> Enrollment Server (U <-> W)

The protocol between U and W is carried between U and V in message_1 and message_2 (Section 4.5), and between V and W in the Voucher Request/Response (Section 4.6). The data is protected between the endpoints using secret keys derived from a Diffie-Hellman shared secret (see Section 4.2) as further detailed in this section.

4.4.1. Voucher Info

The external authorization data EAD_1 contains an EAD item with ead_label = TBD1 and ead_value = Voucher_Info, which is a CBOR byte string:

```
Voucher_Info = bstr .cbor Voucher_Info_Seq

Voucher_Info_Seq = (
    LOC_W:      tstr,
    ENC_ID:     bstr
)
```

where

- LOC_W is a text string used by V to locate W, e.g., a URI or a domain name.
- ENC_ID is a byte string containing an encrypted identifier of U, calculated as follows:

ENC_ID is 'ciphertext' of COSE_Encrypt0 (Section 5.2 of [RFC9052]) computed from the following:

- The encryption key K_1 and nonce IV_1 are derived as specified below.
- 'protected' is a byte string of size 0
- 'plaintext' and 'external_aad' as below:

```
plaintext = (
    ID_U:            bstr,
 )

external_aad = (
    SS:              int,
 )
```

where

- ID_U is an identifier of the device, see Section 3.1.

*SS is the selected cipher suite in SUITES_I of EDHOC message_1, see Section 4.5.

Editor's note: Add more context to external_aad.

The derivation of K_1 = EDHOC-Expand(PRK, info, length) uses the following input to the info struct (see Section 4.2):

*info_label = 0

*context = h'' (the empty CBOR string)

*length is length of key of the EDHOC AEAD algorithm in bytes

The derivation of IV_1 = EDHOC-Expand(PRK, info, length) uses the following input to the info struct (see Section 4.2):

*info_label = 1

*context = h'' (the empty CBOR string)

*length is length of nonce of the EDHOC AEAD algorithm in bytes

4.4.2. Voucher

The voucher is an assertion to U that W has authorized V. The voucher is essentially a message authentication code which binds the authentication credential of V, CRED_V, to message_1 of EDHOC.

The external authorization data EAD_2 contains an EAD item with ead_label = TBD1 and ead_value = Voucher, which is a CBOR byte string:

```
Voucher = bstr .cbor EDHOC-Expand(PRK, info, length)
```

The voucher is calculated with the following input to the info struct (see Section 4.2):

*info_label = 2

*context = bstr .cbor voucher_input

*length is EDHOC MAC length in bytes

where context is a CBOR byte string wrapping of the following CBOR sequence:

```
voucher_input = (
    H(message_1):  bstr,
    CRED_V:        bstr,
)
```

where

- H(message_1) is the hash of EDHOC message_1, calculated from the associated voucher request, see Section 4.6.1.

- CRED_V is the CWT Claims Set [RFC8392] containing the public authentication key of V, see Section 4.5.2.1

4.5. Device <-> Authenticator (U <-> V)

This section describes the processing in U and V, which include the EDHOC protocol, see Figure 3. Normal EDHOC processing is omitted here.

4.5.1. Message 1

4.5.1.1. Processing in U

U composes EDHOC message_1 using authentication method, identifiers, etc. according to an agreed application profile, see Section 3.9 of [I-D.ietf-lake-edhoc]. The selected cipher suite, in this document denoted SS, applies also to the interaction with W as detailed in Section 4.2, in particular, with respect to the Diffie Hellman key agreement algorithm used between U and W. As part of the normal EDHOC processing, U generates the ephemeral public key G_X which is reused in the interaction with W, see Section 4.4.

The device sends EDHOC message_1 with EAD item (-TBD1, Voucher_Info) included in EAD_1, where Voucher_Info is specified in Section 4.4. The negative sign indicates that the EAD item is critical, see Section 3.8 of [I-D.ietf-lake-edhoc].

4.5.1.2. Processing in V

V receives EDHOC message_1 from U and processes it as specified in Section 5.2.3 of [I-D.ietf-lake-edhoc], with the additional step of processing the EAD item in EAD_1. Since the EAD item is critical, if V does not recognize it or it contains information that V cannot process, then V MUST abort the EDHOC session, see Section 3.8 of [I-D.ietf-lake-edhoc]. Otherwise, the ead_label = TBD1, triggers the voucher request to W as described in Section 4.6. The exchange between V and W needs to be completed successfully for the EDHOC session to be continued.

4.5.2. Message 2

4.5.2.1. Processing in V

V receives the voucher response from W as described in Section 4.6.

V sends EDHOC message_2 to U with the critical EAD item (-TBD1, Voucher) included in EAD_2, where the Voucher is specified in Section 4.4.

CRED_V is a CWT Claims Set [RFC8392] containing the public authentication key of V encoded as a COSE_Key in the 'cnf' claim, see Section 3.5.2 of [I-D.ietf-lake-edhoc].

ID_CRED_R contains the CWT Claims Set with 'kccs' as COSE header_map, see Section 9.6 of [I-D.ietf-lake-edhoc].

4.5.2.2. Processing in U

U receives EDHOC message_2 from V and processes it as specified in Section 5.3.2 of [I-D.ietf-lake-edhoc], with the additional step of processing the EAD item in EAD_2.

If U does not recognize the EAD item or the EAD item contains information that U cannot process, then U MUST abort the EDHOC session, see Section 3.8 of [I-D.ietf-lake-edhoc]. Otherwise U MUST verify the Voucher by performing the same calculation as in Section 4.4.2 using H(message_1) and CRED_V received in ID_CRED_R of message_2. If the voucher calculated in this way is not identical to what was received in message_2, then U MUST abort the EDHOC session.

4.5.3. Message 3

4.5.3.1. Processing in U

If all verifications are passed, then U sends EDHOC message_3.

EDHOC message_3 may be combined with an OSCORE request, see [I-D.ietf-core-oscore-edhoc].

4.5.3.2. Processing in V

V performs the normal EDHOC verifications of message_3. V may retrieve CRED_U from a Credential Database, after having learnt ID_CRED_I from U.

4.6. Authenticator <-> Enrollment Server (V <-> W)

It is assumed that V and W have set up a secure connection, W has accessed the authentication credential CRED_V to be used in the

EDHOC session between V and with U, and that W has verified that V is in possession of the private key corresponding to CRED_V, see Section 3.2 and Section 3.3. V and W run the Voucher Request/ Response protocol over the secure connection.

4.6.1. Voucher Request

4.6.1.1. Processing in V

V sends the voucher request to W. The Voucher Request SHALL be a CBOR array as defined below:

```
Voucher_Request = [
    message_1:      bstr,
    ? opaque_state: bstr
]
```

where

- message_1 is the EDHOC message_1 as it was received from U.
- opaque_state is OPTIONAL and represents the serialized and encrypted opaque state needed by V to statelessly respond to U after the reception of Voucher_Response.

4.6.1.2. Processing in W

W receives and parses the voucher request received over the secure connection with V. The voucher request essentially contains EDHOC message_1 as sent by U to V. W SHALL NOT process message_1 as if it was an EDHOC message intended for W.

W extracts from message_1:

- SS - the selected cipher suite, which is the (last) integer of SUITES_I.
- G_X - the ephemeral public key of U
- ENC_ID - the encryption of the device identifier ID_U, contained in the Voucher_Info field of the EAD item with ead_label = TBD1 (with minus sign indicating criticality)

W verifies and decrypts ENC_ID using the relevant algorithms of the selected cipher suite SS (see Section 4.2), and obtains ID_U.

W calculates the hash of message_1 H(message_1), and associates this session identifier to the device identifier ID_U. If H(message_1) is not unique among session identifiers associated to this device identifier of U, the EDHOC session SHALL be aborted.

W uses ID_U to look up the associated authorization policies for U and enforces them. This is out of scope for the specification.

4.6.2. Voucher Response

4.6.2.1. Processing in W

W retrieves CRED_V associated to the secure connection with V, and constructs the the Voucher for the device with identifier ID_U (see Section 4.4.2).

W generates the voucher response and sends it to V over the secure connection. The Voucher_Response SHALL be a CBOR array as defined below:

```
Voucher_Response = [
    message_1:      bstr,
    Voucher:        bstr,
    ? opaque_state: bstr
]
```

where

- *message_1 is the EDHOC message_1 as it was received from V.
- *The Voucher is defined in Section 4.4.2.
- *opaque_state is the echoed byte string opaque_state from Voucher_Request, if present.

4.6.2.2. Processing in V

V receives the voucher response from W over the secure connection. If present, V decrypts and verifies opaque_state as received from W. If that verification fails then EDHOC is aborted. If the voucher response is successfully received from W, then V responds to U with EDHOC message_2 as described in Section 4.5.2.1.

5. REST Interface at W

The interaction between V and W is enabled through a RESTful interface exposed by W. This RESTful interface MAY be implemented using either HTTP or CoAP. V SHOULD access the resources exposed by W through the protocol indicated by the scheme in LOC_W URI.

5.1. Scheme "https"

In case the scheme indicates "https", V MUST perform a TLS handshake with W and use HTTP. If the authentication credential CRED_V can be used in a TLS handshake, e.g. an X.509 certificate of a signature

public key, then V SHOULD use it to authenticate to W as a client. If the authentication credential CRED_V cannot be used in a TLS handshake, e.g. if the public key is a static Diffie-Hellman key, then V SHOULD first perform a TLS handshake with W using available compatible keys. V MUST then perform an EDHOC session over the TLS connection proving to W the possession of the private key corresponding to CRED_V. Performing the EDHOC session is only necessary if V did not authenticate with CRED_V in the TLS handshake with W.

Editor's note: Clarify that performing TLS handshake is not necessary for each device request; if there already is a TLS connection between V and W that should be reused. Similar considerations for 5.2 and 5.3.

5.2. Scheme "coaps"

In case the scheme indicates "coaps", V SHOULD perform a DTLS handshake with W and access the resources defined in Section 5.4 using CoAP. The normative requirements in Section 5.1 on performing the DTLS handshake and EDHOC session remain the same, except that TLS is replaced with DTLS.

5.3. Scheme "coap"

In case the scheme indicates "coap", V SHOULD perform an EDHOC session with W, as specified in Appendix A of [I-D.ietf-lake-edhoc] and access the resources defined in Section 5.4 using OSCORE and CoAP. The authentication credential in this EDHOC session MUST be CRED_V.

5.4. URIs

The URIs defined below are valid for both HTTP and CoAP. W MUST support the use of the path-prefix "/.well-known/", as defined in [RFC8615], and the registered name "lake-authz". A valid URI in case of HTTP thus begins with

* "https://www.example.com/.well-known/lake-authz"

In case of CoAP with DTLS:

* "coaps://example.com/.well-known/lake-authz"

In case of EDHOC and OSCORE:

* "coap://example.com/.well-known/lake-authz"

Each operation specified in the following is indicated by a path-suffix.

5.4.1. Voucher Request (/voucherrequest)

To request a voucher, V MUST issue a request:

- *Method is POST
- *Payload is the serialization of the Voucher Request object, as specified in Section 4.6.1.
- *Content-Format (Content-Type) is set to "application/lake-authz-voucherrequest+cbor"

In case of successful processing at W, W MUST issue a 200 OK response with payload containing the serialized Voucher Response object, as specified in Section 4.6.2.

5.4.2. Certificate Request (/certrequest)

V requests the public key certificate of U from W through the "/certrequest" path-suffix. To request U's authentication credential, V MUST issue a request:

- *Method is POST
- *Payload is the serialization of the ID_CRED_I object, as received in EDHOC message_3.

In case of a successful lookup of the authentication credential at W, W MUST issue 200 OK response with payload containing the serialized CRED_U.

6. Security Considerations

This specification builds on and reuses many of the security constructions of EDHOC, e.g., shared secret calculation and key derivation. The security considerations of EDHOC [I-D.ietf-lake-edhoc] apply with modifications discussed here.

EDHOC provides identity protection of the Initiator, here the device. The encryption of the device identifier ID_U in the first message should consider potential information leaking from the length of ID_U, either by making all identifiers having the same length or the use of a padding scheme.

Although W learns about the identity of U after receiving VREQ, this information must not be disclosed to V, until U has revealed its identity to V with ID_CRED_I in message_3. W may be used for lookup of CRED_U from ID_CRED_I, or this credential lookup function may be separate from the authorization function of W, see Figure 3. The trust model used here is that U decides to which V it reveals its

identity. In an alternative trust model where U trusts W to decide to which V it reveals U's identity, CRED_U could be sent in Voucher Response.

As noted in Section 8.2 of [I-D.ietf-lake-edhoc] an ephemeral key may be used to calculate several ECDH shared secrets. In this specification the ephemeral key G_X is also used to calculate G_XW, the shared secret with the enrollment server.

The private ephemeral key is thus used in the device for calculations of key material relating to both the authenticator and the enrollment server. There are different options for where to implement these calculations, one option is as an addition to EDHOC, i.e., to extend the EDHOC API in the device with input of public key of W (G_W) and device identifier of U (ID_U), and produce the encryption of ID_U which is included in Voucher_Info in EAD_1.

7. IANA Considerations

7.1. EDHOC External Authorization Data Registry

IANA has registered the following entry in the "EDHOC External Authorization Data" registry under the group name "Ephemeral Diffie-Hellman Over COSE (EDHOC)". The ead_label = TBD_1 corresponds to the ead_value Voucher_Info in EAD_1, and Voucher in EAD_2 with processing specified in Section 4.5.1 and Section 4.5.2, respectively, of this document.

Label	Value Type	Description
TBD1	bstr	Voucher related information

Table 1: Addition to the EDHOC EAD registry

7.2. The Well-Known URI Registry

IANA has registered the following entry in "The Well-Known URI Registry", using the template from [RFC8615]:

*URI suffix: lake-authz

*Change controller: IETF

*Specification document: [[this document]]

*Related information: None

7.3. Well-Known Name Under ".arpa" Name Space

This document allocates a well-known name under the .arpa name space according to the rules given in [RFC3172] and [RFC6761]. The name

"lake-authz.arpa" is requested. No subdomains are expected, and addition of any such subdomains requires the publication of an IETF Standards Track RFC. No A, AAAA, or PTR record is requested.

7.4. Media Types Registry

IANA has added the media types "application/lake-authz-voucherrequest+cbor" to the "Media Types" registry.

7.4.1. application/lake-authz-voucherrequest+cbor Media Type Registration

*Type name: application

*Subtype name: lake-authz-voucherrequest+cbor

*Required parameters: N/A

*Optional paramaters: N/A

*Encoding considerations: binary

*Security cosniderations: See Section 6 of this document.

*Interoperability considerations: N/A

*Published specification: [[this document]] (this document)

*Application that use this media type: To be identified

*Fragment identifier considerations: N/A

*Additional information:

  -Magic number(s): N/A

  -File extension(s): N/A

  -Macintosh file type code(s): N/A

*Person & email address to contact for further information: See "Authors' Addresses" section.

*Intended usage: COMMON

*Restrictions on usage: N/A

*Author: See "Authors' Addresses" section.

*Change Controller: IESG

7.5. CoAP Content-Formats Registry

IANA has added the media type "application/lake-authz-voucherrequest+cbor" to the "CoAP Content-Formats" registry under the registry group "Constrained RESTful Environments (CoRE) Parameters".

Media Type	Encoding	ID	Reference
application/lake-authz-voucherrequest+cbor	-	TBD2	[[this document]]

Table 2: Addition to the CoAP Content-Formats registry

8. References

8.1. Normative References

[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-20, 7 July 2023, <https://datatracker.ietf.org/api/v1/doc/document/draft-ietf-lake-edhoc/>.

[NIST-800-56A] Barker, E., Chen, L., Roginsky, A., Vassilev, A., and R. Davis, "Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography - NIST Special Publication 800-56A, Revision 3", April 2018, <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Ar3.pdf>.

[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>.

[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>.

[RFC9052] Schaad, J., "CBOR Object Signing and Encryption (COSE): Structures and Process", STD 96, RFC 9052, DOI 10.17487/RFC9052, August 2022, <https://www.rfc-editor.org/info/rfc9052>.

8.2. Informative References

[I-D.ietf-core-oscore-edhoc] Palombini, F., Tiloca, M., Höglund, R., Hristozov, S., and G. Selander, "Using EDHOC with CoAP and OSCORE", Work in Progress, Internet-Draft, draft-ietf-core-oscore-edhoc-07, 13 March 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-core-oscore-edhoc-07>.

[I-D.ietf-lake-reqs] Vučinić, M., Selander, G., Mattsson, J. P., and D. Garcia-Carillo, "Requirements for a Lightweight AKE for OSCORE", Work in Progress, Internet-Draft, draft-ietf-lake-reqs-04, 8 June 2020, <https://datatracker.ietf.org/doc/html/draft-ietf-lake-reqs-04>.

[IEEE802.15.4] IEEE standard for Information Technology, "IEEE Std 802.15.4 Standard for Low-Rate Wireless Networks", n.d..

[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>.

[RFC3172] Huston, G., Ed., "Management Guidelines & Operational Requirements for the Address and Routing Parameter Area Domain ("arpa")", BCP 52, RFC 3172, DOI 10.17487/RFC3172, September 2001, <https://www.rfc-editor.org/info/rfc3172>.

[RFC6761] Cheshire, S. and M. Krochmal, "Special-Use Domain Names", RFC 6761, DOI 10.17487/RFC6761, February 2013, <https://www.rfc-editor.org/info/rfc6761>.

[RFC7228] Bormann, C., Ersue, M., and A. Keranen, "Terminology for Constrained-Node Networks", RFC 7228, DOI 10.17487/RFC7228, May 2014, <https://www.rfc-editor.org/info/rfc7228>.

[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>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[RFC8615] Nottingham, M., "Well-Known Uniform Resource Identifiers (URIs)", RFC 8615, DOI 10.17487/RFC8615, May 2019, <https://www.rfc-editor.org/info/rfc8615>.

[RFC8995] Pritikin, M., Richardson, M., Eckert, T., Behringer, M., and K. Watsen, "Bootstrapping Remote Secure Key Infrastructure (BRSKI)", RFC 8995, DOI 10.17487/RFC8995, May 2021, <https://www.rfc-editor.org/info/rfc8995>.

[RFC9031] Vučinić, M., Ed., Simon, J., Pister, K., and M. Richardson, "Constrained Join Protocol (CoJP) for 6TiSCH", RFC 9031, DOI 10.17487/RFC9031, May 2021, <https://www.rfc-editor.org/info/rfc9031>.

Appendix A. Use with Constrained Join Protocol (CoJP)

This section outlines how the protocol is used for network enrollment and parameter provisioning. An IEEE 802.15.4 network is used as an example of how a new device (U) can be enrolled into the domain managed by the domain authenticator (V).

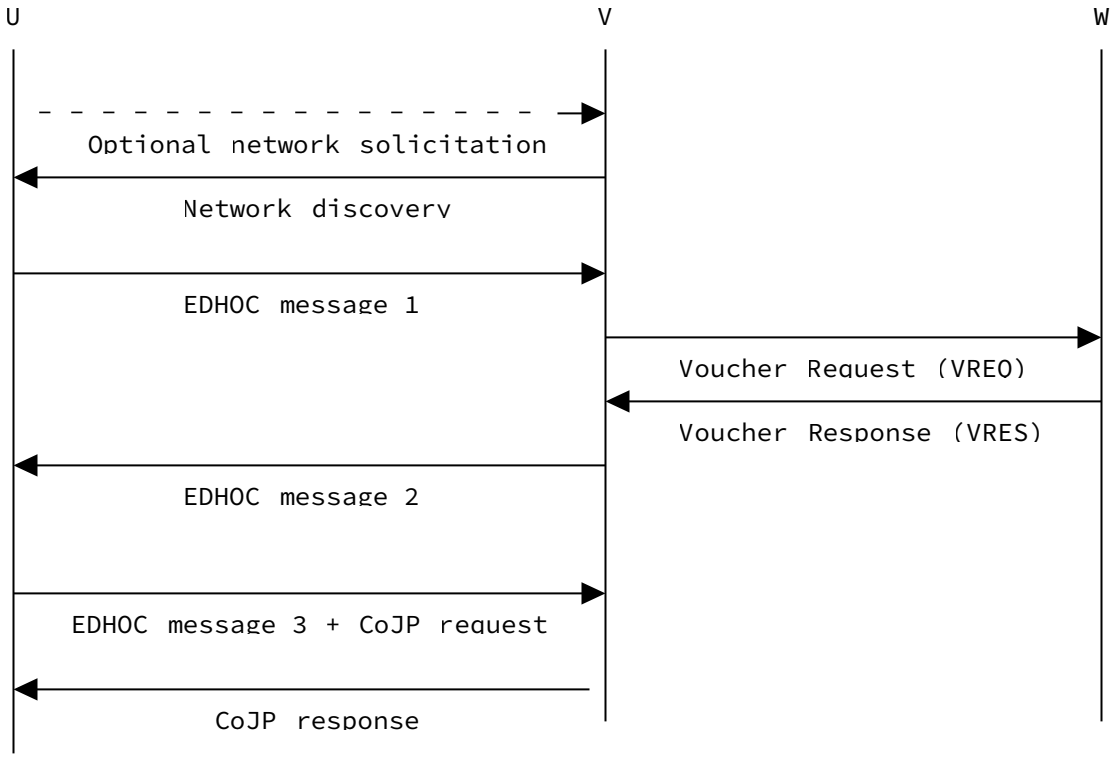

Figure 4: Use of draft-selander-lake-authz with CoJP.

A.1. Network Discovery

When a device first boots, it needs to discover the network it attempts to join. The network discovery procedure is defined by the link-layer technology in use. In case of Time-slotted Channel

Hopping (TSCH) networks, a mode of [IEEE802.15.4], the device scans the radio channels for Enhanced Beacon (EB) frames, a procedure known as passive scan. EBs carry the information about the network, and particularly the network identifier. Based on the EB, the network identifier, the information pre-configured into the device, the device makes the decision on whether it should join the network advertised by the received EB frame. This process is described in Section 4.1 of [RFC9031]. In case of other, non-TSCH modes of IEEE 802.15.4 it is possible to use the active scan procedure and send solicitation frames. These solicitation frames trigger the nearest network coordinator to respond by emitting a beacon frame. The network coordinator emitting beacons may be multiple link-layer hops away from the domain authenticator (V), in which case it plays the role of a Join Proxy (see [RFC9031]). Join Proxy does not participate in the protocol and acts as a transparent router between the device and the domain authenticator. For simplicity, Figure 4 illustrates the case when the device and the domain authenticator are a single hop away and can communicate directly.

A.2. The Enrollment Protocol with Parameter Provisioning

A.2.1. Flight 1

Once the device has discovered the network it wants to join, it constructs the EDHOC message_1, as described in Section 4.5. The device SHALL map the message to a CoAP request:

- *The request method is POST.
- *The type is Confirmable (CON).
- *The Proxy-Scheme option is set to "coap".
- *The Uri-Host option is set to "lake-authz.arpa". This is an anycast type of identifier of the domain authenticator (V) that is resolved to its IPv6 address by the Join Proxy.
- *The Uri-Path option is set to ".well-known/edhoc".
- *The Content-Format option is set to "application/cid-edhoc+cbor-seq"
- *The payload is the (true, EDHOC message_1) CBOR sequence, where EDHOC message_1 is constructed as defined in Section 4.5.

A.2.2. Flight 2

The domain authenticator receives message_1 and processes it as described in Section 4.5. The message triggers the exchange with the enrollment server, as described in Section 4.6. If the exchange

between V and W completes successfully, the domain authenticator prepares EDHOC message_2, as described in Section 4.5. The authenticator SHALL map the message to a CoAP response:

- *The response code is 2.04 Changed.
- *The Content-Format option is set to "application/edhoc+cbor-seq"
- *The payload is the EDHOC message_2, as defined in Section 4.5.

A.2.3. Flight 3

The device receives EDHOC message_2 and processes it as described in Section 4.5}. Upon successful processing of message_2, the device prepares flight 3, which is an OSCORE-protected CoJP request containing an EDHOC message_3, as described in [I-D.ietf-core-oscore-edhoc]. EDHOC message_3 is prepared as described in Section 4.5. The OSCORE-protected payload is the CoJP Join Request object specified in Section 8.4.1 of [RFC9031]. OSCORE protection leverages the OSCORE Security Context derived from the EDHOC session, as specified in Appendix A of [I-D.ietf-lake-edhoc]. To that end, [I-D.ietf-core-oscore-edhoc] specifies that the Sender ID of the client (device) must be set to the connection identifier selected by the domain authenticator, C_R. OSCORE includes the Sender ID as the kid in the OSCORE option. The network identifier in the CoJP Join Request object is set to the network identifier obtained from the network discovery phase. In case of IEEE 802.15.4 networks, this is the PAN ID.

The device SHALL map the message to a CoAP request:

- *The request method is POST.
- *The type is Confirmable (CON).
- *The Proxy-Scheme option is set to "coap".
- *The Uri-Host option is set to "lake-authz.arpa".
- *The Uri-Path option is set to ".well-known/edhoc".
- *The EDHOC option [I-D.ietf-core-oscore-edhoc] is set and is empty.
- *The payload is prepared as described in Section 3.2 of [I-D.ietf-core-oscore-edhoc], with EDHOC message_3 and the CoJP Join Request object as the OSCORE-protected payload.

Note that the OSCORE Sender IDs are derived from the connection identifiers of the EDHOC session. This is in contrast with [RFC9031]

where ID Context of the OSCORE Security Context is set to the device identifier (pledge identifier). Since the device identity is exchanged during the EDHOC session, and the certificate of the device is communicated to the authenticator as part of the Voucher Response message, there is no need to transport the device identity in OSCORE messages. The authenticator playing the role of the [RFC9031] JRC obtains the device identity from the execution of the authorization protocol.

A.2.4. Flight 4

Flight 4 is the OSCORE response carrying CoJP response message. The message is processed as specified in Section 8.4.2 of [RFC9031].

Authors' Addresses

Göran Selander
Ericsson AB
Sweden

Email: goran.selander@ericsson.com

John Preuß Mattsson
Ericsson AB
Sweden

Email: john.mattsson@ericsson.com

Mališa Vučinić
INRIA
France

Email: malisa.vucinic@inria.fr

Michael Richardson
Sandelman Software Works
Canada

Email: mcr+ietf@sandelman.ca

Aurelio Schellenbaum
Institute of Embedded Systems, ZHAW
Switzerland

Email: aureliorubendario.schellenbaum@zhaw.ch