

Lightweight Authenticated Key Exchange  
Internet-Draft  
Intended status: Standards Track  
Expires: 8 January 2026

Y. Song  
Inria  
G. Selander  
Ericsson AB  
7 July 2025

Remote attestation over EDHOC  
draft-ietf-lake-ra-02

## Abstract

This document specifies how to perform remote attestation as part of the lightweight authenticated Diffie-Hellman key exchange protocol EDHOC (Ephemeral Diffie-Hellman Over COSE), based on the Remote ATtestation procedureS (RATS) architecture.

## 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://lake-wg.github.io/ra/draft-ietf-lake-ra.html>. Status information for this document may be found at <https://datatracker.ietf.org/doc/draft-ietf-lake-ra/>.

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/lake-wg/ra>.

## 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 2026.

## Copyright Notice

Copyright (c) 2025 IETF Trust and the persons identified as the document authors. All rights reserved.

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

## Table of Contents

1. Introduction . . . . .	3
2. Conventions and Definitions . . . . .	4
3. Overview . . . . .	5
4. Assumptions . . . . .	5
5. Remote Attestation in EDHOC . . . . .	6
5.1. Target: IoT Device Attestation (IoT) . . . . .	6
5.2. Target: Network Service Attestation (Net) . . . . .	6
5.3. Model: Background-check Model (BG) . . . . .	6
5.3.1. Attestation_proposal . . . . .	7
5.3.2. Attestation_request . . . . .	8
5.3.3. Evidence . . . . .	9
5.3.4. trigger_bg . . . . .	10
5.4. Model: Passport Model (PP) . . . . .	10
5.4.1. Result_proposal . . . . .	11
5.4.2. Result_request . . . . .	11
5.4.3. Result . . . . .	12
5.4.4. trigger_pp . . . . .	12
5.5. EDHOC forward Message Flow (Fwd) . . . . .	12
5.6. EDHOC reverse Message Flow (Rev) . . . . .	12
6. Instantiation of Remote Attestation Protocol . . . . .	13
6.1. (IoT, BG, Fwd): IoT Device Attestation . . . . .	13
6.2. (Net, PP, Fwd): Network Service Attestation . . . . .	15
7. Mutual Attestation in EDHOC . . . . .	17
7.1. (IoT, BG, Fwd) - (Net, PP, Fwd) . . . . .	17
8. Verifier . . . . .	19

8.1. Processing in the Background-check Model . . . . .	19
8.2. Processing in the Passport Model . . . . .	20
9. Security Considerations . . . . .	20
10. IANA Considerations . . . . .	21
10.1. EDHOC External Authorization Data Registry . . . . .	21
11. References . . . . .	21
11.1. Normative References . . . . .	21
11.2. Informative References . . . . .	22
Appendix A. Example: Remote Attestation Flow . . . . .	23
Appendix B. Remote attestation in parallel with enrollment authorization . . . . .	24
Appendix C. Example: Firmware Version . . . . .	24
Appendix D. Open discussion: Continuous Attestation over EDHOC . . . . .	26
Acknowledgments . . . . .	27
Authors' Addresses . . . . .	27

## 1. Introduction

Remote attestation is a security process which verifies and confirms the integrity and trustworthiness of a remote device or system in the network. This process helps establish a level of trust in the remote system before allowing the device to e.g. join the network or get access to some sensitive information and resources. The use cases that require remote attestation include secure boot and firmware management, cloud computing, network access control, etc.

The IETF working group Remote ATtestation procedures (RATS) has defined an architecture [RFC9334] for remote attestation. The three main roles in the RATS architecture are the Attester, the Verifier and the Relying Party. The Attester generates the evidence concerning its identity and integrity, which must be appraised by the Verifier for its validity. Then, the Verifier produces the attestation result, which is consequently used by the Relying Party for the purposes of reliably applying application-specific actions.

One type of interaction model defined in the RATS architecture is called the background-check model. It resembles the procedure of how employers perform background checks to determine the prospective employee's trustworthiness, by contacting the respective organization that issues a report. In this case, the employer acts as the Relying Party, the employee acts as the Attester and the organization acts as the Verifier. The Attester conveys evidence directly to the Relying Party and the Relying Party forwards the evidence to the Verifier for appraisal. Once the attestation result is computed by the Verifier, it is sent back to the Relying Party to decide what action to take based on the attestation result. Another model is called passport model, where the Attester communicates directly with the Verifier.

The Attester presents the evidence to the Verifier and gets an attestation result from the Verifier. Then the Attester conveys the attestation result to the Relying Party. This specification employs both the RATS background-check model and the passport model.

This document specifies the protocol between the Attester and the Relying Party. The details of the protocol between the Relying Party and the Verifier in the background-check model, and the protocol between the Attester and the Verifier in the passport model are out of the scope. This communication may be secured through protocols such as EDHOC, TLS or other security protocols that support the secure transmission to and from the Verifier.

One way of conveying attestation evidence or the attestation result is the Entity Attestation Token (EAT) [RFC9711]. It provides an attested claims set which can be used to determine a level of trustworthiness. This specification relies on the EAT as the format for attestation evidence and the attestation result.

Ephemeral Diffie-Hellman over COSE (EDHOC) [RFC9528] is a lightweight authenticated key exchange protocol for highly constrained networks. In EDHOC, the two parties involved in the key exchange are referred to as the Initiator (I) and the Responder (R). EDHOC supports the transport of external authorization data, through the dedicated EAD fields. This specification delivers EAT through EDHOC. Specifically, EAT is transported as an EAD item. This specification also defines new EAD items needed to perform remote attestation over EDHOC in Section 5.3 and Section 5.4.

For the generation of evidence, the Attester incorporates an internal attestation service, including a specific trusted element known as the "root of trust". Root of trust serves as the starting point for establishing and validating the trustworthiness appraisals of other components on the system. The measurements signed by the attestation service are referred to as the Evidence. The signing is requested through an attestation API. How the components are separated between the secure and non-secure worlds on a device is out of scope of this specification.

## 2. Conventions and Definitions

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 assumed to be familiar with the terms and concepts defined in EDHOC [RFC9528] and RATS architecture [RFC9334].

### 3. Overview

This specification describes how to perform remote attestation over the EDHOC protocol, following the RATS architecture. EDHOC provides the benefit of minimal message overhead and reduced round trips for a lightweight authentication. More importantly, by integrating remote attestation with EDHOC, attestation can be run in parallel with authentication, improving the efficiency and maintaining lightweight properties.

Remote attestation protocol elements are carried within EDHOC's External Authorization Data (EAD) fields. EDHOC [RFC9528] supports one or more EAD items in each EAD field.

In Section 5, this document defines three independent dimensions for performing remote attestation over EDHOC:

1. Target (see Section 5.1, Section 5.2) defining the entity that undergoes the attestation process (IoT device or network service).
2. Model (see Section 5.3, Section 5.4) defining the attestation model in use based on the RATS architecture (background-check model or passport model).
3. Message Flow (see Section 5.5, Section 5.6) defining the EDHOC message flow in use (forward message flow or reverse message flow).

This document specifies the cases that are suited for constrained IoT environments.

### 4. Assumptions

In the background-check model, the Verifier is assumed to support verification of at least one evidence format provided by the Attester. The Verifier is assumed to be provisioned with the Attester's attestation public key and the reference values required for evidence validation prior to the attestation procedure. It is assumed that the Relying Party also has knowledge about the Attester, so it can narrow down the evidence type selection and send to the Attester only one format of the evidence type.

In the passport model, the credential identity of the Verifier is assumed to be stored at the Attester and the Relying Party, which means the Verifier is trusted by the Attester and the Relying Party to obtain the attestation result. If timestamps are used to ensure freshness in the passport model, synchronized time between the Attester and Relying Party is assumed. For detailed time considerations, refer to Appendix A of [RFC9334].

## 5. Remote Attestation in EDHOC

This section specifies three independent dimensions that characterize the remote attestation process over EDHOC.

1. Target: Defines the entity that undergoes the attestation process.
2. Model: Defines the attestation models based on RATS architecture. This specification supports both the RATS background-check model (see Section 5.3) and the passport model (see Section 5.4). The corresponding EAD items for background-check model and the passport model are independent of each other.
3. EDHOC message flow: The EDHOC protocol defines two possible message flows, namely the EDHOC forward message flow and the EDHOC reverse message flow (see Appendix A.2.2 of [RFC9528]). In this specification, both flows can be used to perform remote attestation.

### 5.1. Target: IoT Device Attestation (IoT)

The IoT device acts as the Attester.

### 5.2. Target: Network Service Attestation (Net)

The network service acts as the Attester.

This attestation pattern applies when constrained devices need to establish trust with a network gateway. For example, before transmitting sensitive data to a network gateway, a constrained device requires attestation of the network gateway.

### 5.3. Model: Background-check Model (BG)

In the background-check model, the Attester sends the evidence to the Relying Party. The Relying Party transfers the evidence to the Verifier and gets back the attestation result from the Verifier.

An EDHOC session is established between the Attester and the Relying Party. A negotiation of the evidence type is required before the Attester sends the evidence. All three parties must agree on a selected evidence type.

The Attester first sends a list of the proposed evidence types to the Relying Party. The list is formatted as an Attestation proposal in an EDHOC EAD item. The Relying Party relays the list to the Verifier and receives at least one supported evidence type from the Verifier in return. If the Relying Party receives more than one evidence type, a single evidence type should be selected by the Relying Party based on its knowledge of the Attester. The Relying Party then sends it back within the Attestation request to the Attester.

A nonce, at least 8-bytes long [RFC9711]), guarantees the freshness of the attestation session. The nonce is generated by the Verifier and sent to the Relying Party. The Relying Party puts the nonce and the selected evidence type together in a tuple to generate an Attestation request.

Once the Attester receives the Attestation request, it can call its attestation service to generate the evidence, with the nonce value as one of the inputs.

The `ead_label` for `Attestation_proposal`, `Attestation_request` and `Evidence` is the same (TBD1) because these EAD items appear in a fixed sequential order.

#### 5.3.1. `Attestation_proposal`

To propose a list of provided evidence types in background-check model, the Attester transports the `Proposed_EvidenceType` object. It signals to the Relying Party the proposal to do remote attestation, as well as which types of the attestation claims the Attester supports. The `Proposed_EvidenceType` is encoded in CBOR in the form of a sequence.

The EAD item for an attestation proposal is:

- \* `ead_label` = TBD1

- \* `ead_value` = `Attestation_proposal`, which is a CBOR byte string:

```
Attestation_proposal = bstr .cbor Proposed_EvidenceType
```

```
Proposed_EvidenceType = [ + content-format ]
```

```
content-format = uint
```

where

- \* Proposed\_EvidenceType is an array that contains all the supported evidence types by the Attester.
- \* There MUST be at least one item in the array.
- \* content-format is an indicator of the format type (e.g., application/eat+cwt with an appropriate eat\_profile parameter set), from [IANA-CoAP-Content-Formats].

The sign of ead\_label TBD1 MUST be negative to indicate that the EAD item is critical. If the receiver cannot recognize the critical EAD item, or cannot process the information in the critical EAD item, then the receiver MUST abort the EDHOC session, as defined in Section 3.8 of [RFC9528].

### 5.3.2. Attestation\_request

As a response to the attestation proposal, the Relying Party signals to the Attester the supported and requested evidence type. In case none of the evidence types is supported, the Relying Party rejects the first message\_1 with an error indicating support for another evidence type.

The EAD item for an attestation request is:

- \* ead\_label = TBD1
- \* ead\_value = Attestation\_request, which is a CBOR byte string:

Attestation\_request = bstr .cbor Selected\_EvidenceType

```
Selected_EvidenceType = (  
  content-format: uint,  
  nonce: bstr .size 8..64  
)
```

where

- \* content-format is the selected evidence type by the Relying Party and supported by the Verifier.
- \* nonce is generated by the Verifier and forwarded by the Relying Party.



The sign of ead\_label TBD2 MUST be negative to indicate that the EAD item is critical. If the receiver cannot recognize the critical EAD item, or cannot process the information in the critical EAD item, then the receiver MUST abort the EDHOC session, as defined in Section 3.8 of [RFC9528].

### 5.3.3. Evidence

The Attester calls its local attestation service to generate and return a serialized Entity Attestation Token (EAT) [RFC9711] as Evidence. The Evidence is sent in EAD\_3 within EDHOC message\_3.

The EAD item is:

- \* ead\_label = TBD1
- \* ead\_value is a serialized EAT.

For remote attestation over EDHOC, The EAT MUST be formatted as a CBOR Web Token (CWT) containing attestation-oriented claims. The complete set of attestation claims for the EAT is specified in [RFC9711].

A minimal claims set MAY be implemented when the Attester operates under constrained message size requirements and/or limited computational resources.

```
{
/eat-nonce/          10: bstr .size 8
/ueid/               256: bstr .size (7..33)
/measurements/       273: measurements-type
}
```

The measurements claim MAY be formatted according to [RFC9393]. When CoSWID [RFC9393] is used, the claim MUST be an evidence CoSWID rather than a payload CoSWID. Formats other than CoSWID are permitted and MUST be identified by CoAP Content Format.

measurements-type = [+ measurements-format]

```
measurements-format = [
    content-format: uint,
    content: bytes
]
```

#### 5.3.4. trigger\_bg

The EAD item `trigger_bg` is used when the sender triggers the receiver to start a remote attestation in the background-check model. The receiver **MUST** reply with an EAD item corresponding to the background-check model, either an `attestation_proposal` in Section 5.3.1 or an evidence in Section 5.3.3. The `ead_value` **MUST** not be present, as the `ead_label` serves as the trigger.

The EAD item is:

- \* `ead_label` = TBD2
- \* `ead_value` = null

#### 5.4. Model: Passport Model (PP)

In the passport model, the Attester sends the evidence to the Verifier. After the Attester receives the attestation result from the Verifier, the Attester sends the attestation result to the Relying Party.

An EDHOC session is established between the Attester and the Relying Party. The Attester and the Relying Party should decide from which Verifier the Attester obtains the attestation result and transfers it to the Relying Party. The Attester first sends a list of the Verifier identities that it can get the attestation result. The Relying Party selects one trusted Verifier identity and sends it back as a Result request.

Regarding the freshness in passport model, the Attester could either establish a real-time connection with the selected Verifier, or use a pre-stored attestation result from the selected Verifier. If the attestation result is not obtained via a real-time connection, it should include a time stamp and/or expiry time to indicate its validity. Time synchronization is out of scope of this specification.

Once the Attester obtains the attestation result from the selected Verifier, it sends the attestation result to the Relying Party.

The `ead_label` for `Result_proposal`, `Result_request` and `Result` is the same (TBD3) because these EAD items appear in a fixed sequential order.

#### 5.4.1. Result\_proposal

An attestation result proposal contains the identification of the credentials of the Verifiers to indicate Verifiers' identities. The identification of credentials relies on COSE header parameters [IANA-COSE-Header-Parameters], with a header label and credential value.

The EAD item for the attestation result proposal is:

- \* ead\_label = TBD3

- \* ead\_value = Result\_proposal, which is a CBOR byte string:

```
Result_proposal = bstr .cbor Proposed_VerifierIdentity
Proposed_VerifierIdentity = [ + VerifierIdentity ]
```

```
VerifierIdentity = {
  label => values
}
```

where

- \* Proposed\_VerifierIdentity is defined as a list of one or more VerifierIdentity elements.

- \* Each VerifierIdentity within the list is a map defined in [IANA-COSE-Header-Parameters] that:

- label = int / tstr
- values = any

#### 5.4.2. Result\_request

As a response to the attestation result proposal, the Relying Party signals to the Attester the trusted Verifier. In case none of the Verifiers can be trusted by the Relying Party, the session is aborted. Relying Party generates a nonce to ensure the freshness of the attestation result from the Verifier.

The EAD item for an attestation result request is:

- \* ead\_label = TBD3

- \* ead\_value = Result\_request, which is a CBOR byte string:

```
Result_request = bstr .cbor Request_structure
```

```
Request_structure = {  
  selected_verifier: VerfierIdentity,  
  ? nonce: bstr .size 8..64  
}
```

#### 5.4.3. Result

The attestation result is generated and signed by the Verifier as a serialized EAT [RFC9711]. The Relying Party can decide what action to take with regards to the Attester based on the information elements in attestation result.

The EAD item is:

- \* ead\_label = TBD3
- \* ead\_value is a serialized EAT.

#### 5.4.4. trigger\_pp

The EAD item trigger\_pp is used when the sender triggers the receiver to start a remote attestation in the passport model. The receiver MUST reply with an EAD item correspondign to the passport model, either a result proposal in Section 5.4.1 or a result in Section 5.4.3. The ead\_value MUST not be present, as the ead\_label serves as the trigger.

The EAD item is:

- \* ead\_label = TBD4
- \* ead\_value = null

#### 5.5. EDHOC forward Message Flow (Fwd)

In forward message flow, EDHOC may run with the Initiator as a CoAP/HTTP client. Remote attestation over EDHOC starts with a POST requests to the Uri-Path: `"/.well-known/lake-ra"`.

#### 5.6. EDHOC reverse Message Flow (Rev)

In the reverse message flow, the CoAP/HTTP client is the Responder and the server is the Initiator. A new EDHOC session begins with an empty POST request to the server's resource for EDHOC.

## 6. Instantiation of Remote Attestation Protocol

We use the format (Target, Model, Message Flow) to denote instantiations. For example, (IoT, BG, Fwd) represents IoT device attestation using the background-check model with forward EDHOC message flow.

There are 8 possible instantiations combining IoT/Net, BG/PP, and Fwd/Rev configurations. Considering practical feasibility, IoT device attestation in the passport model and Network Service attestation in the background model are less viable. These instantiations require IoT devices to maintain multiple network connections and perform complex computations, which exceed the capabilities of typical low-power, resource-constrained IoT devices.

In this document, only the most relevant instantiations for constrained IoT environments are specified.

### 6.1. (IoT, BG, Fwd): IoT Device Attestation

A common use case for (IoT, BG, Fwd) is to attest an IoT device to a network server. For example, a simple illustrative case is doing remote attestation to verify that the latest version of firmware is running on the IoT device before the network server allows it to join the network (see Appendix C).

An overview of doing IoT device attestation in background-check model and EDHOC forward message flow is established in Figure 1. EDHOC Initiator plays the role of the RATS Attester (A). EDHOC Responder plays the role of the RATS Relying Party (RP). The Attester and the Relying Party communicate by transporting messages within EDHOC's External Authorization Data (EAD) fields. An external entity, out of scope of this specification, plays the role of the RATS Verifier (V). The EAD items specific to the background-check model are defined in Section 5.3.

The Attester starts the attestation by sending an Attestation proposal in EDHOC message\_1. The Relying Party generates EAD\_2 with the received evidence type and nonce from the Verifier, and sends it to the Attester. The Attester sends the Evidence to the Relying Party in EAD\_3. The Verifier evaluates the Evidence and sends the Attestation result to the Relying Party.

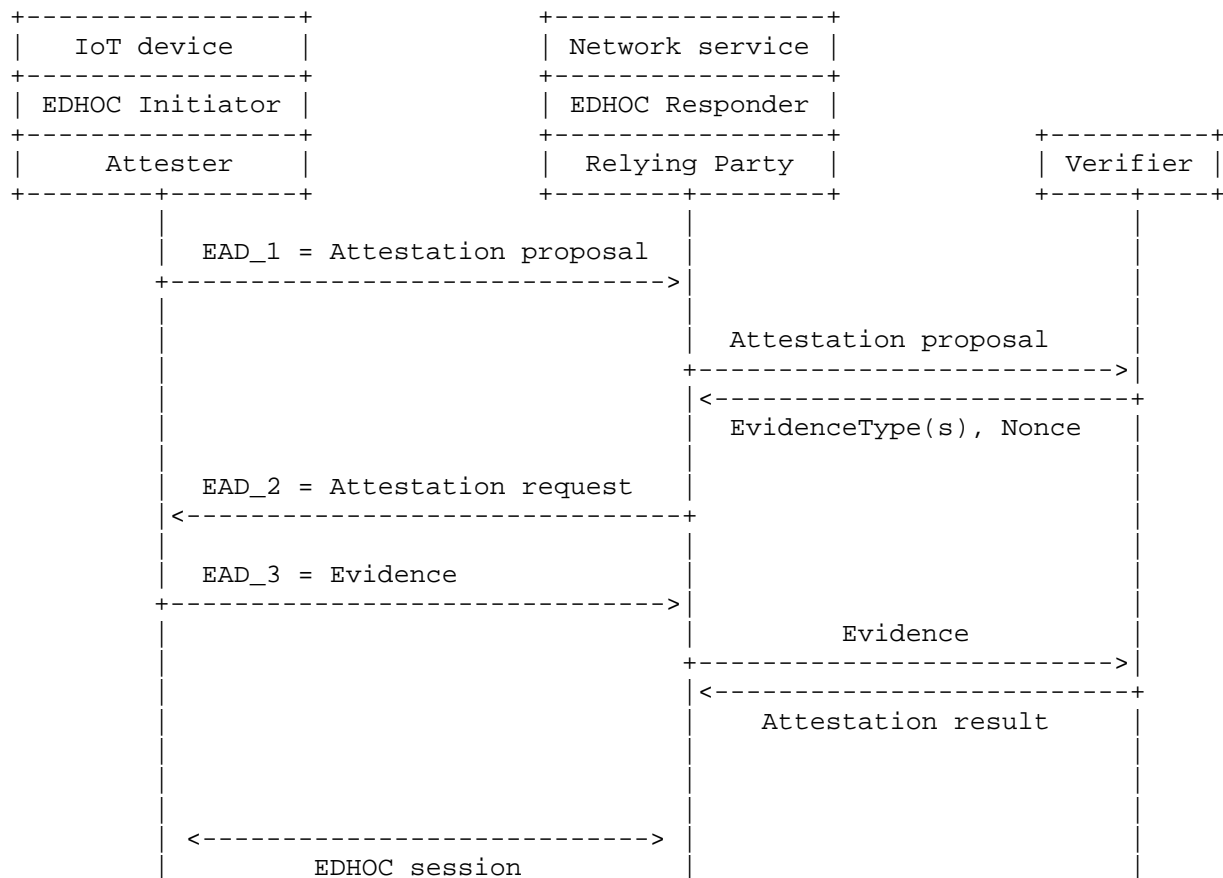


Figure 1: Overview of IoT device attestation in background-check model and EDHOC forward message flow. EDHOC is used between A and RP.

This specification also supports a simplified attestation model for scenarios where a pre-existing relationship exists between the Attester and the Verifier. In such cases, when the Verifier possesses prior knowledge of the evidence format (CBOR) and its expected profile of the EAT claims set (e.g., CoSWID), the evidence types negotiation MAY be omitted. Under these circumstances, the attestation process is simplified as:

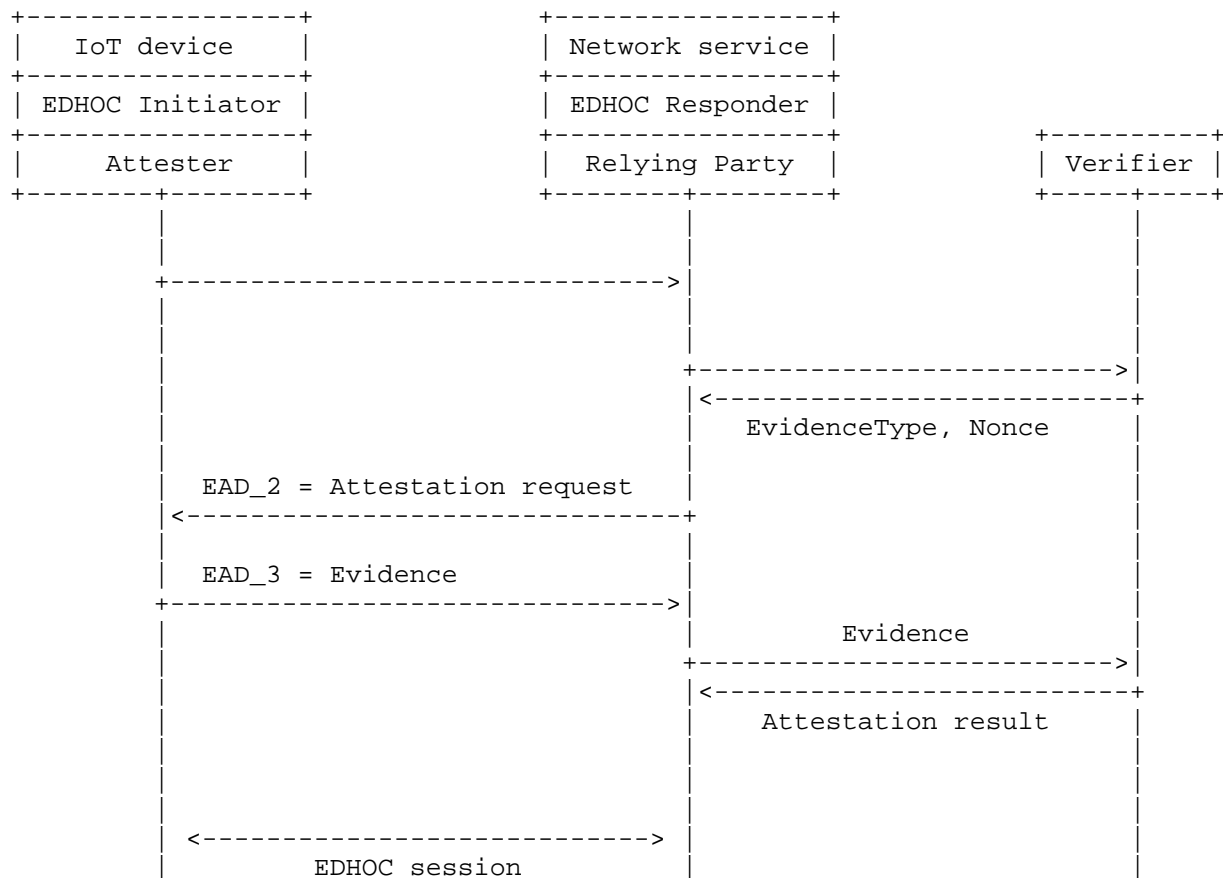


Figure 2: Simplified IoT device attestation in background-check model with EDHOC forward message flow. Pre-existing relationship between A and V.

## 6.2. (Net, PP, Fwd): Network Service Attestation

One use case for (Net, PP, Fwd) is when a network server needs to attest itself to a client (e.g., an IoT device). For example, the client needs to send some sensitive data to the network server, which requires the network server to be attested first. In (Net, PP, Fwd), the network server acts as an Attester and the client acts as a Relying Party.

An overview of the message flow is illustrated in Figure 3. EDHOC Initiator plays the role of the RATS Relying Party. EDHOC Responder plays the role of the RATS Attester. An external entity, out of scope of this specification, plays the role of the RATS Verifier. The EAD items specific to the passport model are defined in Section 5.4.

The Relying Party asks the Attester to do a remote attestation by sending a `trigger_pp` (see Section 5.4.4) in EDHOC `message_1`. The Attester replies to the Relying Party with a Result proposal in `EAD_2`. Then the Relying Party selects a trusted Verifier identity and sends it as a Result request. How the Attester negotiates with the selected Verifier to get the attestation result is out of scope of this specification. A fourth EDHOC message is required to send the Result from the Attester to the Relying Party.

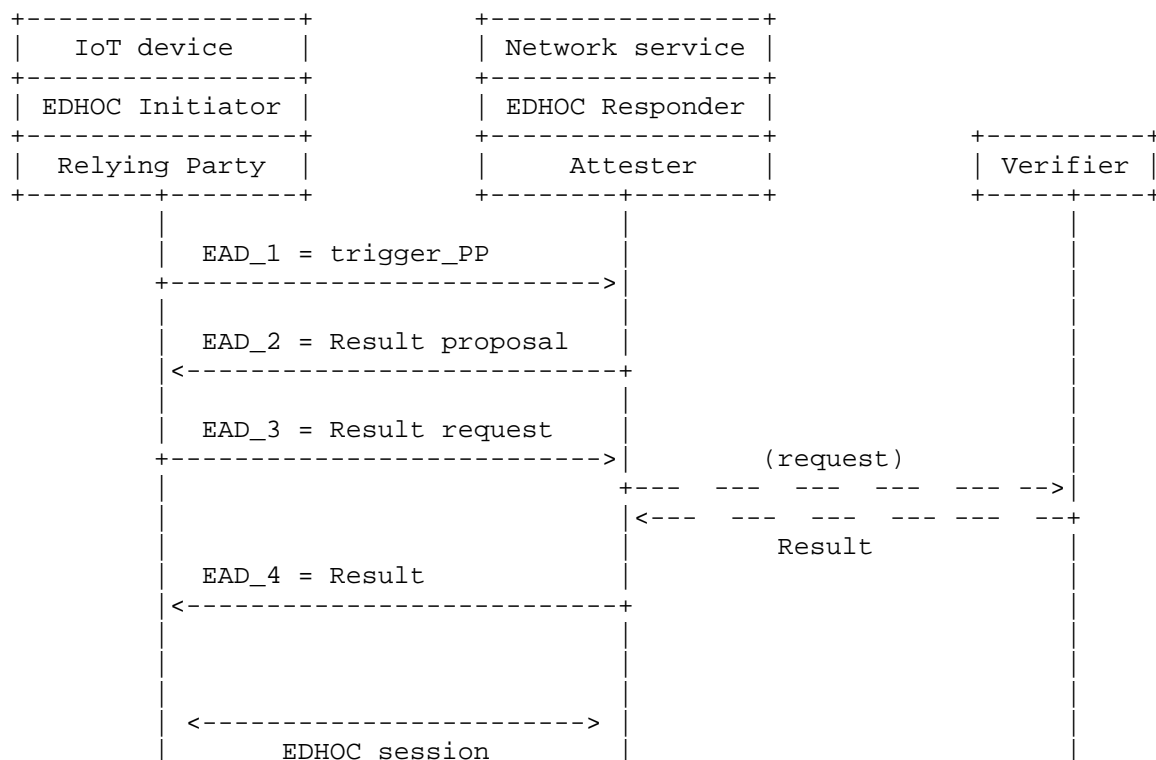


Figure 3: Overview of network service attestation in passport model and EDHOC forward message flow. EDHOC is used between RP and A. The dashed line illustrates a logical connection that does not need to occur in real time.



## 7. Mutual Attestation in EDHOC

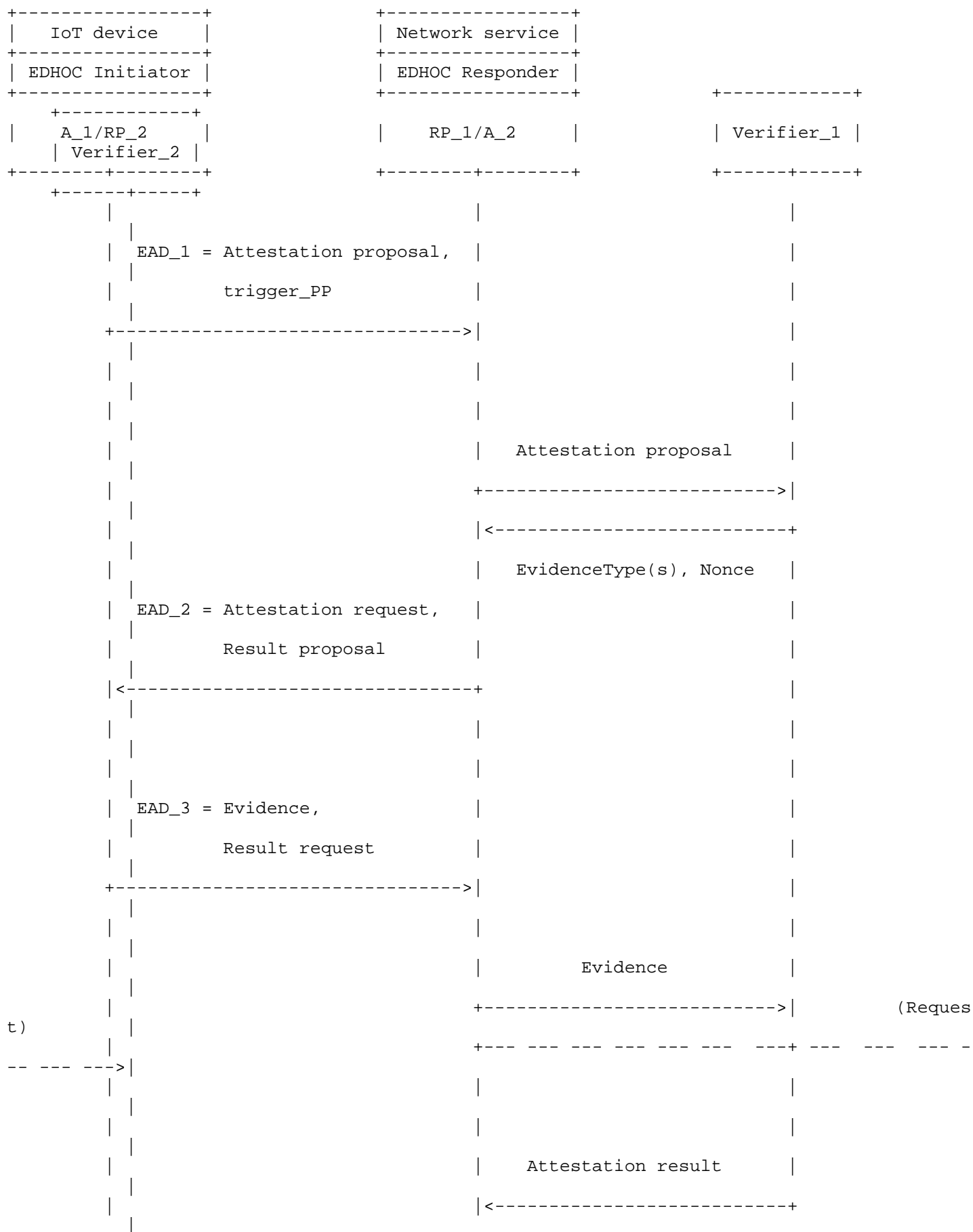
Mutual attestation over EDHOC combines the cases where one of the EDHOC parties uses the IoT device attestation and the other uses the Network service attestation. Performing mutual attestation to a single EDHOC message flow achieves a lightweight use with reduced message overhead. Note that the message flow for the two parties in mutual attestation needs to be the same.

In this specification, we list the most relevant mutual attestation example for constrained IoT environments.

### 7.1. (IoT, BG, Fwd) - (Net, PP, Fwd)

In this example, the mutual attestation is performed in EDHOC forward message flow, by one IoT device attestation in background-check model and another network service attestation in passport model. The process is illustrated in Figure 4. How the Network service connects with the Verifier\_1 and potential Verifier\_2 is out of scope in this specification.

The first remote attestation is initiated by the IoT device (A\_1) in background-check model. In parallel, the IoT device (A\_1) requests the network service (A\_2) to perform a remote attestation in passport model. EAD\_2 carries the EAD items Attestation request and Result proposal. EAD\_3 carries the EAD items Evidence and Result request. EAD\_4 carries the EAD item Result for the passport model.



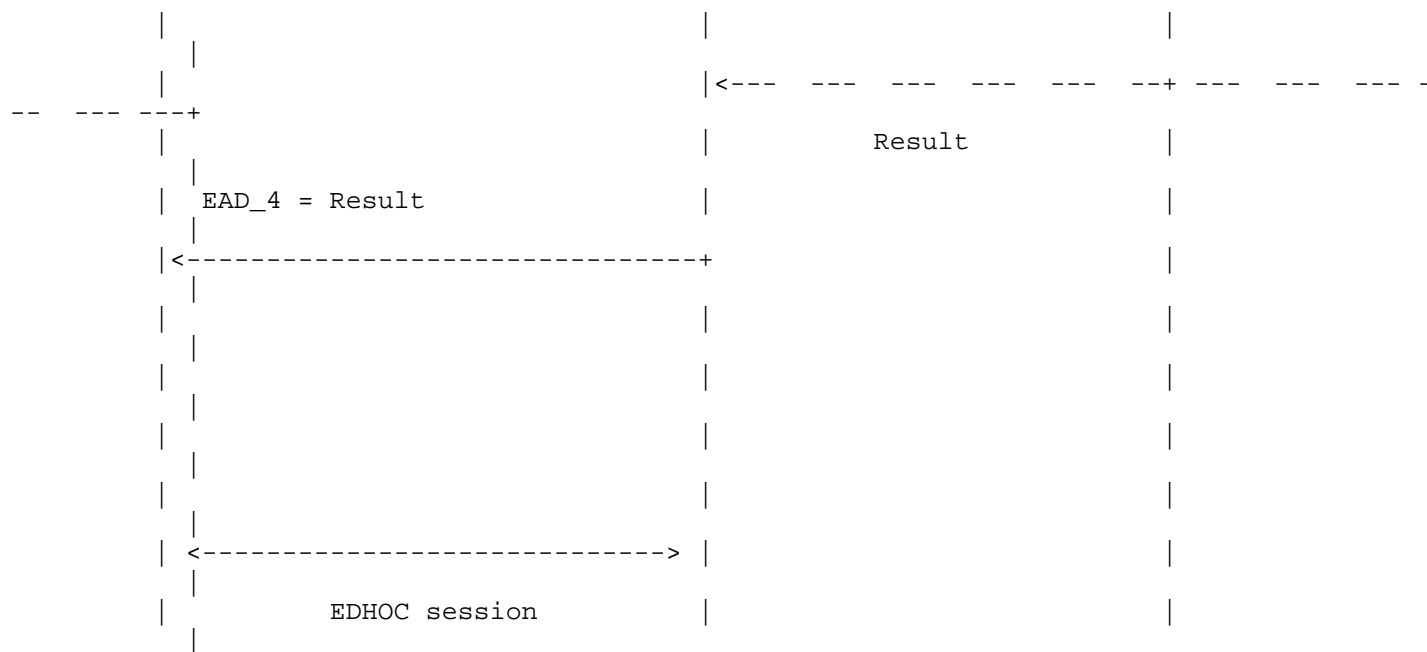


Figure 4: Overview of mutual attestation of (IoT, BG, Fwd) - (Net, PP, Fwd). EDHOC is used between A and RP. The dashed line illustrates a logical connection that does not need to occur in real time.

## 8. Verifier

The Verifier maintains an explicit trust relationship with the Attester, whereby the Verifier is provisioned with the Attester's attestation public key prior to the remote attestation process. This explicit relationship may be established through various means, such as manufacturer provisioning, trusted certification authorities, or direct configuration. Reference values used for comparison against received evidence should also be provided to the Verifier before the attestation. The evaluation policy employed by the Verifier varies according to specific use cases and should be determined prior to the attestation; such policy definition is out of scope in this specification.

The Verifier maintains an implicit trust relationship with the Relying Party, established through mechanisms such as web PKI with trusted Certificate Authority (CA) certificates, enabling the Relying Party to trust the attestation result that is generated by the Verifier.

### 8.1. Processing in the Background-check Model

The Verifier is connected with the Relying Party and is responsible for evaluating evidence forwarded by the Relying Party. After the Relying Party receives EDHOC message\_1 from the Attester, it extracts and transmits the Attestation proposal to the Verifier. The Verifier must support at least one evidence type for evaluation, otherwise it returns an empty list. Alongside the selected evidence type, the Verifier generates a random nonce and sends both elements to the Relying Party.

When the Relying Party obtains EDHOC message\_3, it forwards the evidence to the Verifier for evaluation.

The evidence evaluation process should include the verification of the signature, nonce validation, and measurement value comparison. An example evaluation procedure for evidence formatted as an Entity Attestation Token (EAT) within a COSE\_Sign1 structure is as follows:

1. Decode the COSE\_Sign1 structure and extract constituent components: Headers, Payload, Signature.
2. Verify the signature using the Attester's attestation public key.

3. Verify that the nonce exists in the Verifier's local nonce list. If the nonce is found, validation passes and the nonce is removed from the list to prevent replay attacks.
4. Compare the received evidence measurement values against the reference value. The attestation result is returned to the Relying Party, with result generation conforming to the attestation token format defined in [RFC9711].

## 8.2. Processing in the Passport Model

When the Attester utilizes a cached attestation result previously generated by the Verifier, real-time re-evaluation by the Verifier is not required. If the Attester performs real-time attestation with the Verifier after receiving a `result_request` from the Relying Party, the Verifier generates the attestation result formatted as an Entity Attestation Token (EAT). The token uses the "Software Measurement Results (measres)" claim as defined in [RFC9711], and incorporates the nonce generated by the Relying Party as an input parameter.

## 9. Security Considerations

This specification is performed over EDHOC [RFC9528] by using EDHOC's EAD fields. The privacy considerations of EADs in EDHOC apply to this specification.

EAD\_1 is not resistant to either active attackers or passive attackers, because neither the Initiator nor the Responder has been authenticated.

Although EAD\_2 is encrypted, the Initiator has not been authenticated, rendering EAD\_2 vulnerable against the active attackers.

The ead items in EAD\_1 and EAD\_2 MAY be very specific and potentially reveal sensitive information about the device. The leaking of the data in EAD\_1 and/or EAD\_2 MAY risk to be used by the attackers for malicious purposes. Data in EAD\_3 and EAD\_4 are protected between the Initiator and the Responder in EDHOC.

Mutual attestation carries a lower risk for EAD items when the Responder is the Attester. For the mutual attestation at the EDHOC Responder, only the `Attestation_proposal/Result_proposal` in EAD\_2 is not protected to active attackers. Both the `Attestation_request/Result_request` in EAD\_3 and the `Evidence/Result` in EAD\_4 are protected.

## 10. IANA Considerations

### 10.1. EDHOC External Authorization Data Registry

IANA is requested to register the following entry in the "EDHOC External Authorization Data" registry under the group name "Ephemeral Diffie-Hellman Over Cose (EDHOC)".

- \* The ead\_label = TBD1 corresponds to the ead\_value Attestation\_proposal in Section 5.3.1, Attestation\_request in Section 5.3.2 and Evidence in Section 5.3.3.
- \* The ead\_label = TBD2 corresponds to the EAT item trigger\_bg as specified in Section 5.3.4.
- \* The ead\_label = TBD3 corresponds to the ead\_value Result\_proposal in Section 5.4.1, Result\_request in Section 5.4.2 and the Result in Section 5.4.3.
- \* The ead\_label = TBD4 corresponds to the EAT item trigger\_pp as specified in Section 5.4.4.

Name	Label	Description	Reference
TBD	TBD1	BG model related information	Section 5.2.1.1
TBD	TBD2	trigger to start attestation in BG	Section 5.2.1.1
TBD	TBD3	PP model related information	Section 5.2.2.1
TBD	TBD4	trigger to start attestation in PP	Section 5.2.2.1

Figure 5: EAD labels.

## 11. References

### 11.1. Normative References

- [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/rfc/rfc2119>>.

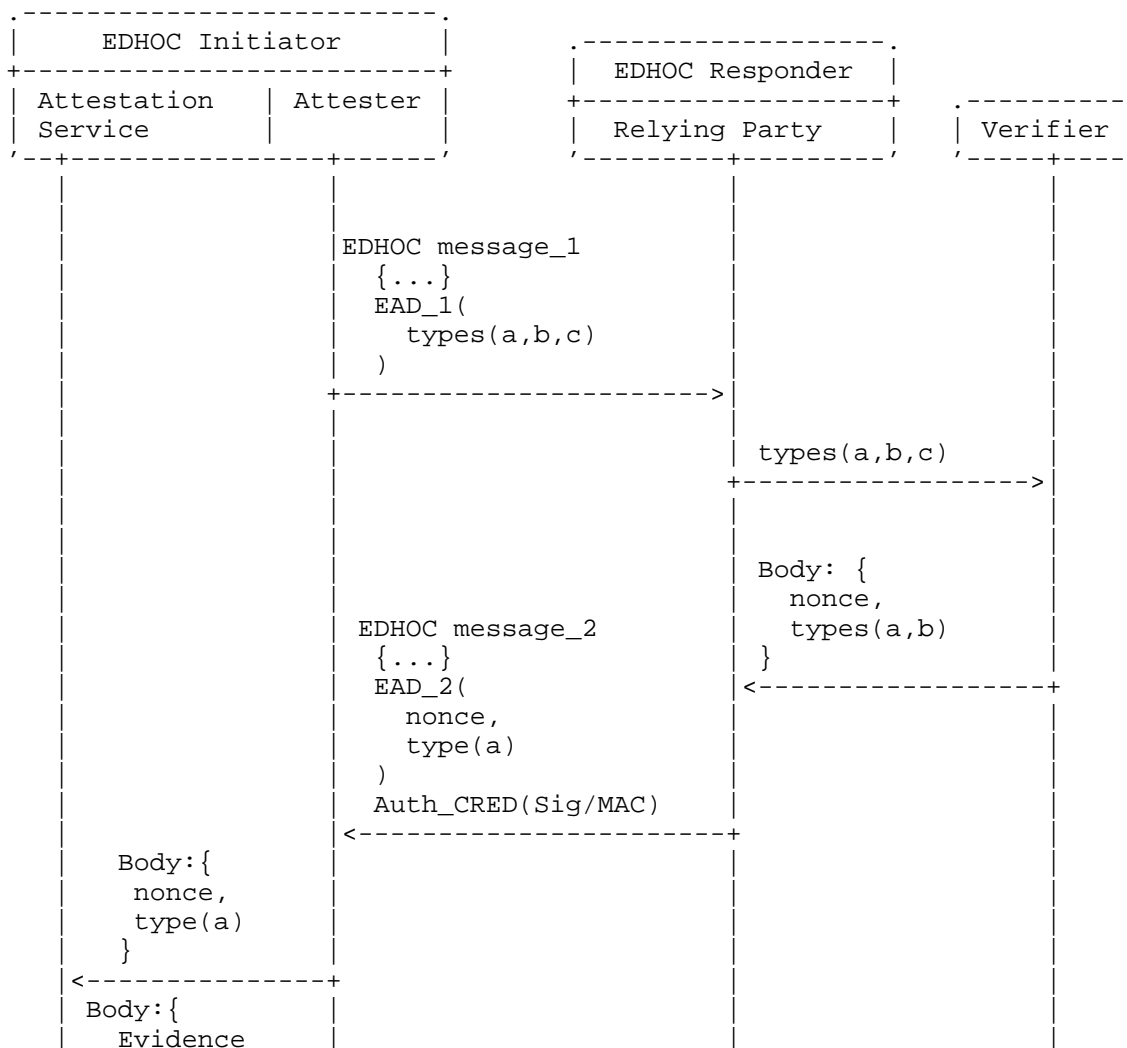
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/rfc/rfc8174>>.
- [RFC9528] Selander, G., Preu Mattsson, J., and F. Palombini, "Ephemeral Diffie-Hellman Over COSE (EDHOC)", RFC 9528, DOI 10.17487/RFC9528, March 2024, <<https://www.rfc-editor.org/rfc/rfc9528>>.
- [RFC9711] Lundblade, L., Mandyam, G., O'Donoghue, J., and C. Wallace, "The Entity Attestation Token (EAT)", RFC 9711, DOI 10.17487/RFC9711, April 2025, <<https://www.rfc-editor.org/rfc/rfc9711>>.

## 11.2. Informative References

- [I-D.ietf-lake-authz]  
Selander, G., Mattsson, J. P., Vuini, M., Fedrecheski, G., and M. Richardson, "Lightweight Authorization using Ephemeral Diffie-Hellman Over COSE (ELA)", Work in Progress, Internet-Draft, draft-ietf-lake-authz-05, 7 July 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-lake-authz-05>>.
- [IANA-CoAP-Content-Formats]  
"CoAP Content-Formats", n.d., <<https://www.iana.org/assignments/core-parameters>>.
- [IANA-COSE-Header-Parameters]  
"COSE Header Parameters", n.d., <<https://www.iana.org/cose/header-parameters>>.
- [IANA.CWT.Claims]  
IANA, "CBOR Web Token (CWT) Claims", <<https://www.iana.org/assignments/cwt>>.
- [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/rfc/rfc8392>>.
- [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/rfc/rfc8949>>.

- [RFC9334] Birkholz, H., Thaler, D., Richardson, M., Smith, N., and W. Pan, "Remote ATtestation procedureS (RATS) Architecture", RFC 9334, DOI 10.17487/RFC9334, January 2023, <<https://www.rfc-editor.org/rfc/rfc9334>>.
- [RFC9393] Birkholz, H., Fitzgerald-McKay, J., Schmidt, C., and D. Waltermire, "Concise Software Identification Tags", RFC 9393, DOI 10.17487/RFC9393, June 2023, <<https://www.rfc-editor.org/rfc/rfc9393>>.

#### Appendix A. Example: Remote Attestation Flow





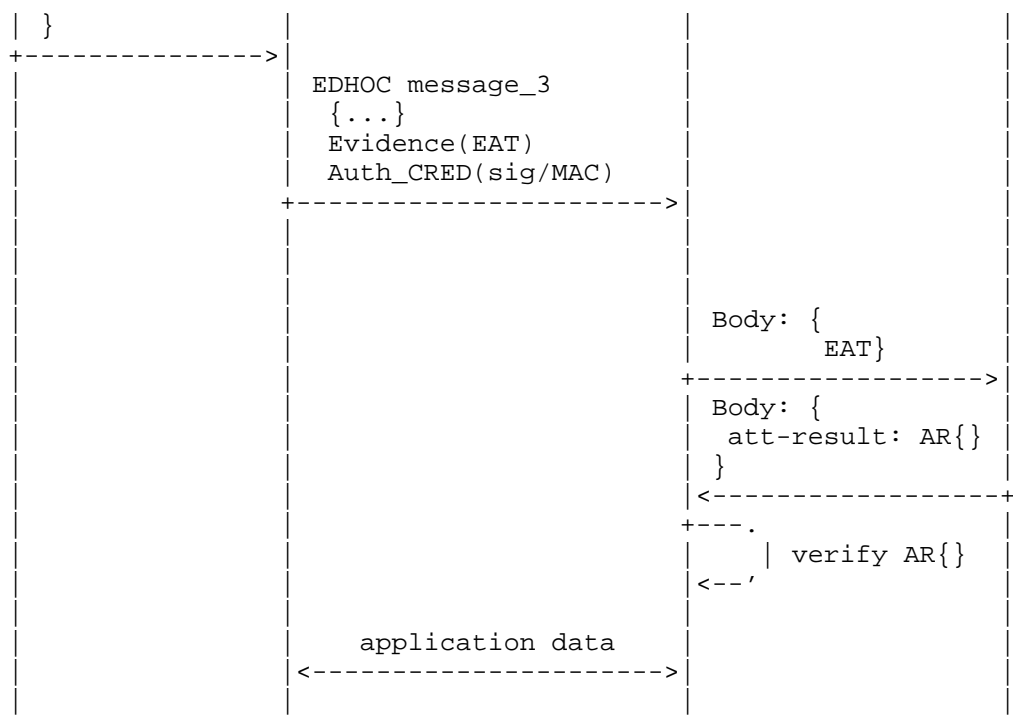


Figure 6: Example of remote attestation.

Appendix B. Remote attestation in parallel with enrollment authorization

This section discusses the possibility of doing remote attestation in parallel with the enrollment authorization procedure defined in [I-D.ietf-lake-authz]. In this case, the message count is much decreased.

The detailed procedure is TBD.

Appendix C. Example: Firmware Version

The goal in this example is to verify that the firmware running on the device is the latest version, and is neither tampered or compromised. A device acts as the Attester, currently in an untrusted state. The Attester needs to generate the evidence to attest itself. A gateway that can communicate with the Attester and can control its access to the network acts as the Relying Party. The gateway will finally decide whether the device can join the network or not depending on the attestation result. The attestation result is produced by the Verifier, which is a web server that can be seen

as the manufacturer of the device. Therefore it can appraise the evidence that is sent by the Attester. The remote attestation session starts with the Attester sending EAD\_1 in EDHOC message 1.

An example of the EAD\_1 in EDHOC message\_1 could be:

```
[60,61,258]
```

If the Verifier and the Relying Party can support at least one evidence type that is proposed by the Attester, the Relying Party will include in the EAD\_2 field the same evidence type, alongside a nonce for message freshness.

```
(258, h'a29f62a4c6cdaae5')
```

The Evidence in EAD\_3 field is an Entity Attestation Token (EAT) [RFC9711], with the measurements claim formatted in CoSWID[RFC9393]. The components of the Evidence should at least be:

```
{
  /eat-nonce/          10: h'a29f62a4c6cdaae5',
  /ueid/               256: 'aaabbcc',
  /measurements/      273: [
    /CoAP Content-Format ID/    [ 258,
    /evidence in CoSWID/        {
      0: 'tagID'                /tag-id/
      12: 0                     /tag-version/
      1: "DotBot firmware"      /software-name/
      2: {                      /entity/
        31: "Attester"         /entity-name/
        33: 1                   /role, must be "tag-creator"
      },
      3: {                      /evidence/
        17: [                  /file/
          {
            24: "partition0-nrf52840dk.bin", /fs-na
me/
of file/
HA-256/
c025bc8b5abca42a19ea0ec23e81a'
value/
            7: [                /hash
              1,                /alg S
              h'06294f6806b9c685eea795048579cfd02a0
              ]                  /hash
            }
          ]
        }
      }
    ]
  ]
}
```

The information above serves as the payload of the COSE object. The complete resulting COSE object is:

```
18([
  /*protected header*/
  h'a10127',

  /*unprotected header*/
  {},

  /*payload*/
  h'A30A48A29F62A4C6CDAAE519010047616161626263631901118182190102A50045746
  16749440C00016F446F74426F74206669726D7761726502A2181F684174746573746572
  18210103A11181A218187819706172746974696F6E302D6E72663532383430646B2E626
  96E078201582006294F6806B9C685EEA795048579CFD02A0C025BC8B5ABCA42A19EA0EC
  23E81A',

  /*signature*/
  h'd4100901f4c3e51312c3110c6ddc8dcf7f68d8f5d3791c19133f2f0ac158c1f5ee6ed
  afe9d7c3d6eb3d2d197f82e733d375fdda9fb258b304961dfc38558950d'
])
```

which has the following base16 encoding:

```
D28443A10127A05890A30A48A29F62A4C6CDAAE51901004761616162626363190111818
2190102A5004574616749440C00016F446F74426F74206669726D7761726502A2181F68
417474657374657218210103A11181A218187819706172746974696F6E302D6E7266353
2383430646B2E62696E078201582006294F6806B9C685EEA795048579CFD02A0C025BC8
B5ABCA42A19EA0EC23E81A5840D4100901F4C3E51312C3110C6DDC8DCF7F68D8F5D3791
C19133F2F0AC158C1F5EE6EDAFE9D7C3D6EB3D2D197F82E733D375FDDA9FB258B304961
DFC38558950D
```

The Relying Party (co-located with the gateway) then treats the Evidence as opaque and sends it to the Verifier. Once the Verifier sends back the Attestation Result, the Relying Party can be assured on the version of the firmware that the device is running.

#### Appendix D. Open discussion: Continuous Attestation over EDHOC

Subsequent remote attestation is possible over PSK-based EDHOC for continuous assurance.

The detailed procedure is TBD.

## Acknowledgments

The author would like to thank Thomas Fossati, Malisa Vucinic, Ionut Mihalcea, Muhammad Usama Sardar, Michael Richardson and Geovane Fedrecheski for the provided ideas and feedback.

Work on this document has in part been supported by the Horizon Europe Framework Programme project OpenSwarm (grant agreement No. 101093046).

## Authors' Addresses

Yuxuan Song  
Inria  
Email: yuxuan.song@inria.fr

Gran Selander  
Ericsson AB  
Email: goran.selander@ericsson.com