Network Working Group
Internet-Draft
Intended status: Experimental
Expires: February 6, 2021

E. Ingles
University of Murcia
D. Garcia
Odin Solutions S.L.
R. Marin
University of Murcia

August 5, 2020

AAA-based assisted EDHOC Authentication
draft-ingles-radex-radius-edhoc-00

Abstract

This document describes a proposal to place EDHOC server in an external Authentication, Authorization and Accounting (AAA) server. The purpose is to centralize the EDHOC authentication in AAA infrastructure.

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 February 6, 2021.

Table of Contents

1. Introduction

EDHOC [I-D.selander-lake-edhoc] is a new protocol for autentication and key derivation that has been proposed as an alternative in IoT mainly due to two main characteristics, namely, it works on top of any reliable transport, which means it can be carried over a protocol such as CoAP and provides a secure exchange in an end-to-end fashion. This key material can be futher used to run other protocols such as OSCORE, as well as providing key material to any other protocol that needs pre-shared key material to secure the communications. EDHOC has another characteristic that makes it an interesting alternative that is work underlining, it is designed to be lightweight, for which is build using COSE, reducing the overhead of the protocol. The proposal of this protocol coalesces with the advancement of the new set of technologies known as LPWAN, which generally have high constrains in the link, even more than traditional IoT networks. Furthermore, these technologies generally lack in measurements for refreshing the key material that is used to protect the communications, for which methods to provide them with bootstrapping and key management capabilities has been subject of reseach, as well

as extending the protocols provided to perform the joining of the devices into the network. In this work we propose an architecture to allow the EDHOC authentication being carried out with the assistance of a AAA infraestructure. The motivation is to centralize not only authentication but also authorization and accounting of a joining IoT node to a particular domain.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. EDHOC support in AAA

Regarding the overall functionality, AAA support for EDHOC will take care of adapting AAA protocols, such as RADIUS or Diameter, to add the support for EDHOC. An example of this could be with RADIUS. In this instance, RADIUS support for EDHOC will define the new Attributes needed to manage the protocol. The EDHOC server implements the RADIUS client supporting this specification and therefore, it MUST implement the RADIUS attributes for this service. The NAS-Port-Type specifying the type of port on which the EDHOC Server is authenticating the End-Device will be set according to the technology used. For example, if we use LoRaWAN [LoRaWAN] we MAY set it to 18 (Wireless - Other) or a new one specifically assigned for LoRaWAN (TBD.). Similarly, the adaptation could be done for Diameter.

3. EDHOC Overview

3.1. Introduction

EDHOC is a lightweight authenticated key exchange protocol that enables to establish a cryptographic key between two entities. To this end, EDHOC implements the Elliptic Curve Diffie-Hellman algorithm with ephemeral keys (ECDHE), by which both entities must generate a new ephemeral key pair every time they launch this protocol. Therefore, EDHOC also provides the perfect forward secrecy property. Additionally, EDHOC supports the same authentication modes as DTLS (i.e., PSK, RPK, and certificates). Hence, the {key generation} process remains independent concerning the selected authentication mode. The EDHOC protocol defines a three-message exchange. These messages are encoded following the CBOR representation and are protected using COSE standard. This way, the minimum message size is assured in contrast to other JSON-based representation formats (such as JWS and JWE), therefore, reducing the overhead on the network. EDHOC protects specific fields selectively

using COSE objects, ensuring end-to-end security, while intermediate entities can access the information required to carry out their functionality.

3.2. EDHOC protocol overview

The EDHOC specification defines an exchange of three messages. The exact message content differs depending, if the authentication method used is PSK, or of RPK or Certificates. The EDHOC client starts the exchange sending the first message that includes the ephimeral public key of the client. When this message arrives to the EDHOC server, it generates it own ephemeral key pair and sends its public key to the client in the second message. The client, then finishes the EDHOC exchange sending the third message.

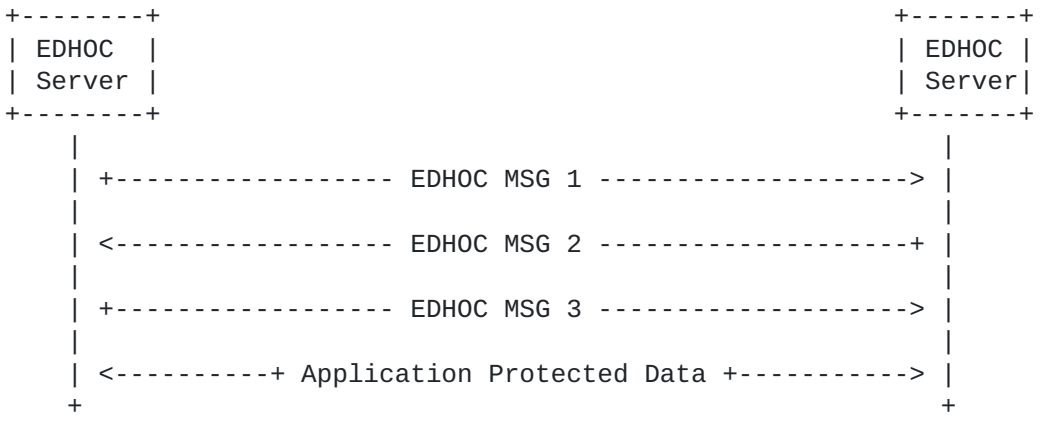


Figure 1: Overview EDHOC exchange

Each message of the EDHOC protocol is defined as a COSE object with specific content depending on the message and the mode of authentication, as specified in the document.

3.3. EDHOC key derivation

When the first message is received by the EDHOC server, it generates its own ephemeral key pair and is able to compute the Secret, called pseudorandom keys (PRK), as follows:

PRK = HKDF-Extract(salt, IKM)

Upon receiving the last exchange both entites have a shared secret key that is derived using a HDKF the input keying material (IKM): Secret, Salt, Context and Key Length. The derivation is done in a different way depending on the method used for authentication.

- The Secret is the same, since it is the result of the Ephemeral Diffie-Hellman exchange as specified above. The Context.

- In case the it is done using PSK, the Salt is the PSK value, otherwise the field is not used.

- The context is the COSE_KDF_CONTEXT defined in the protocol

- The key length is the lenght of the derived shared symetric key that has to be at least 128-bits long.

According to the last version of the draft, there is a key derivation hierarchy by which a Pseudorandom Key (PRK) is derived from the ECDH shared secrets, and from the RPK additional key material called output keying material (OKM) can be also derived.

4. Integration Overview

4.1. Mapping EDHOC entities to AAA infrastructure

In the current specification of EDHOC, there is no explicit reference to an external entity to which the EDHOC Server can degate the authentication. In this sense, we propose to add the support for RADIUS to provide such delegation

4.2. Assumptions

For the integration of EDHOC with RADIUS next we describe some assumptions. The first is that the credentials that are used for authenticating the devices are only shared (in the case of PSK) between the AAA server and the EDHOC client. The outcome of the successful authentication (i.e. PRK) is sent from the AAA server to the EDHOC server. This allows for the EDHOC client to exchange messages with the EDHOC server, once the protocol is finished.

4.3. Protocol Exchange

The join procedure between the client and the server consists if three messages. In RADIUS the EDHOC server implements a RADIUS client to communicate with the AAA server.

The protocol exchange is done in the following steps:

1. The client sends the first message to the EDHOC server.

2. Upon reception of this message, the EDHOC server creates a RADIUS Access-Request message, with the EDHOC-message attribute containing all the fields of the first message of EDHOC.

3. Once the AAA server receives this message, performs the processing of said message as the EDHOC server would in the specification, generating in turn the message 2 and sendig it in a new RADIUS Attribute EDHOC-message, embedded in an Access-Challenge.

4. The EDHOC client, then processes the message and generates the third EDHOC message.

5. The AAA server, receives the third EDHOC message and processes it, deriving the PRK and generating and Access-Accept for the EDHOC server that contains the key in an EDHOC-key attribute.

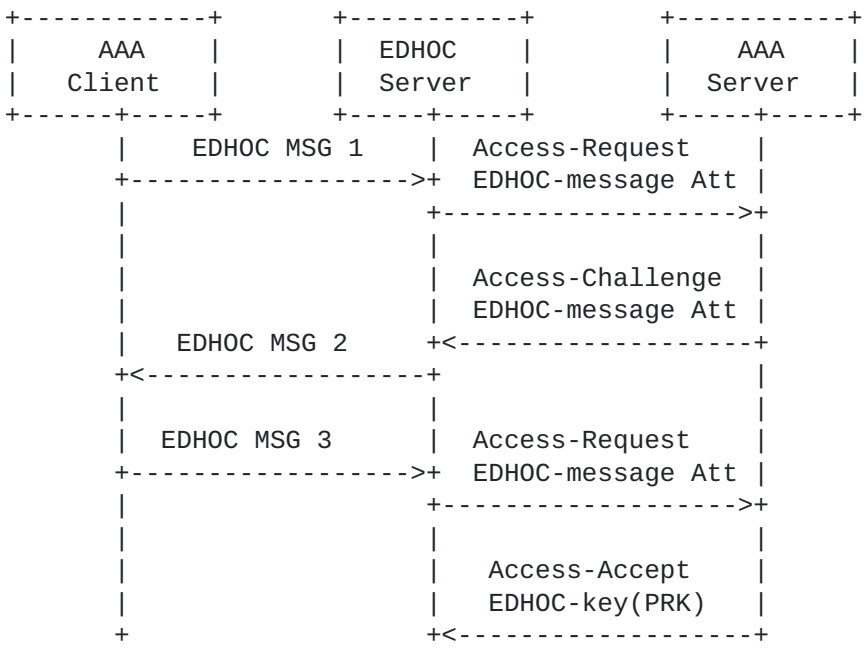
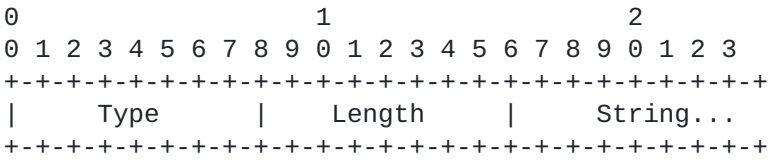


Figure 2: EDHOC-AAA exchange

4.4. EDHOC-message Attribute

Description

This Attribute contains the original EDHOC messages. This attribute will appear in the Access-Request and Access-Challenge messages. A summary of the EDHOC-message attribute format is shown below. The fields are transmitted from left to right.



Type

TBD. for EDHOC-message

Length

>= 3

String

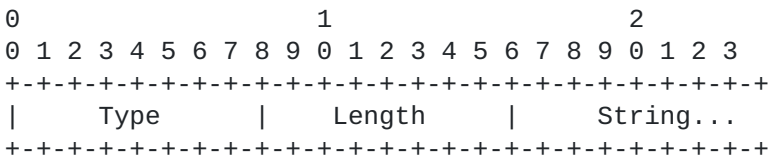
The String field contains an EDHOC message.

The String field contains an octet string with the Join-Request message as received over the network, such as defined in [LoRaWAN].

4.5. EDHOC-key attribute

Description

This Attribute contains the EDHOC PRK, a shared secret key specific for the EDHOC client. This attribute only appears in the RADIUS Access-Accept message. A summary of the EDHOC-key attribute format is shown below. The fields are transmitted from left to right.



Type

TBD. for EDHOC-key

Length

>= 16

String
The String field contains an EDHOC shared symmetric key.

4.6. Table of Attribute

Request	Accept	Reject	Challenge	#	Attribute
1	0	0	1	TBD.	EDHOC-message
0	1	0	0	TBD.	EDHOC-key
Request	Accept	Reject	Challenge	#	Attribute

Figure 3: Attributes Table

5. Open Issues

A specification can be considered about the way credentials are identified in EDHOC to support federation. According to the EDHOC draft, the credentials are identified by each communication endpoing by a 'kid'. We propose that this value will contain a network access identifier, that will be used to retreive the credentials in both the symmetric asymmetric keys. This value is extracted, it is passed to a textual form to include it in an AAA attribute (e.g. User-Name in RADIUS) to be routed to the appropiate server.

6. Security Considerations

The security considerations of this proposal inherit the same security considerations of EDHOC. TBD.

7. Acknowledgements

This work is possible due the EU Project IoTCrawlwer under grant agreement n. 779852 and to the pre-doctoral grant Industrial PhD DI-16-08432 granted to ODIN Solutions S.L

8. IANA Considerations

In this document we define 2 new RADIUS Attributes that would need actions from IANA to assign the corresponding numbers.

Number	Name	Reference
TBD	EDHOC-message	Section 4 of this document
TBD	EDHOC-key	Section 4 of this document

9. References

9.1. Normative References

[I-D.selander-lake-edhoc]
Selander, G., Mattsson, J., and F. Palombini, "Ephemeral Diffie-Hellman Over COSE (EDHOC)", draft-selander-lake-edhoc-01 (work in progress), March 2020.

[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>.

9.2. Informative References

[LoRaWAN] Sornin, N., Luis, M., Eirich, T., and T. Kramp, "LoRa Specification V1.0", January 2015, <https://www.lora-alliance.org/portals/0/specs/LoRaWAN%20Specification%201R0.pdf>.

Authors' Addresses

Eduardo Ingles Sanchez
University of Murcia
Campus de Espinardo S/N, Faculty of Computer Science
Murcia 30100
Spain

Email: eduardo.ingles@um.es

Dan Garcia Carrillo
Odin Solutions S.L.
Poligono Industrial Oeste, C/ Peru, 5
Alcantarilla, Murcia 30820
Spain

Email: dgarcia@odins.es

Rafael Marin-Lopez
University of Murcia
Campus de Espinardo S/N, Faculty of Computer Science
Murcia 30100
Spain

Phone: +34 868 88 85 01
Email: rafa@um.es