**IEEE P802.15**

**Wireless Specialty Networks**

|  |  |  |
| --- | --- | --- |
| Project | IEEE P802.15 Working Group for Wireless Specialty Networks (WSNs) | |
| Title | **TG 802.15.9a List of changes to the IEEE Std 802.15.9** | |
| Date Submitted | 10th September 2024 | |
| Source | Tero Kivinen | E-Mail: kivinen@iki.fi |
| Abstract | List of changes needed in the IEEE Std 802.15.9 to add EDHOC KMP. | |
| Purpose | Getting ready for draft proposal. | |
| Notice | This document has been prepared to assist the IEEE P802.15. It is offered as a basis for discussion and is not binding on the contributing individual(s) or organization(s). The material in this document is subject to change in form and content after further study. The contributor(s) reserve(s) the right to add, amend, or withdraw material contained herein. | |
| Release | The contributor acknowledges and accepts that this contribution becomes the property of IEEE and may be made publicly available by P802.15. | |

1. Changes to Clause 2

Add EDHOC to Acronyms.

1. Changes to Clause 8.1

Add EDHOC algorithm to Table 22.

1. New annex Ea for specifying how to use EDHOC KMP

The new annex would specify how to use EDHOC in the IEEE Std 802.15.9.

Ea.1 Description

EDHOC is a lightweight security handshake protocol standardized by the IETF (RFC 9528). EDHOC provides mutual authentication, ephemeral Diffie-Hellman, forward secrecy, identity protection, and cipher suite negotiation. EDHOC has been analyzed by a wide community of security protocol experts in several studies as part of the design process [[Bruni18](https://www.rfc-editor.org/rfc/rfc9528.html" \l "Bruni18)] [[Norrman20](https://www.rfc-editor.org/rfc/rfc9528.html" \l "Norrman20)] [[CottierPointcheval22](https://www.rfc-editor.org/rfc/rfc9528.html" \l "CottierPointcheval22)] [[GuentherIlunga22](https://www.rfc-editor.org/rfc/rfc9528.html" \l "GuentherIlunga22)] [[Jacomme23](https://www.rfc-editor.org/rfc/rfc9528.html" \l "Jacomme23)] [Ferreira24]

EDHOC enables a low complexity implementation with few and short messages using a generic compact encoding (CBOR, RFC 8949) and security encapsulation (COSE, RFC 9052) reducing message transmission and processing, making it suitable for low-cost / low-power deployments. In one example with pre-provisioned public keys the sum of sizes of the protocol messages is about 100 bytes (see Use cases below).

EDHOC allows different methods for performing mutual authentication, for example using digital signatures or static Diffie-Hellman keys, with the authentication credentials passed by value or by reference. A successful EDHOC session enables the peers to compute an exporter function to derive keys from a shared secret for establishing a security association.

The basic EDHOC protocol has three mandatory and one optional fourth message, and can be combined with OSCORE (RFC 8613) for optimized access to protected REST resources already after a first round trip of EDHOC (RFC 9668).

Ea.2 Use cases

EDHOC is designed to enable a security handshake between two peers potentially running on constrained devices over low-power IoT radio communication technologies, including cellular IoT and other low-power wide area technologies and multi-hop mesh networking. The protocol is beneficial, in particular, with technologies which get performance penalties for large messages in terms of message fragmentation, delays, increased communication costs or reduced battery life.

Small message sizes can be achieved by small authentication credentials passed by value or small identifiers of credentials passed by reference. In one example of a three-message public key based EDHOC session including cryptographic agility and extensibility, each protocol message is smaller than 46 bytes, and the session consists of 101 bytes in total. A detailed trace is shown in RFC 9529.

The protocol can be used for mutual authentication and key agreement between nodes or between node and the coordinator. It can be used in settings where there is no back-end AS, or in an environment where there is a Public Key Infrastructure where certificates are issued.

EDHOC supports authentication methods based on public keys for all credential types and identifiers supported by COSE, including X.509 certificates, C509 certificates [draft-ietf-cose-cbor-encoded-certs], CBOR Web Tokens (CWT, RFC 8392) and CWT Claims Sets (CCS; working as raw public keys), all of the above by value or by reference.

Ea.3 EDHOC and IEEE 802.15 specifics

Ea.3.1 Overview

EDHOC is not bound to a particular transport. EDHOC messages can be transported, for example, directly via KMP payloads or with CoAP as default transport, as specified in RFC 9528.

Ea.3.2 Message framing

Section 3.4 in RFC 9528 lists requirements for transport of EHDOC. Fragmentation and reassembly, and demultiplexing can be provided by IEEE 802.15.9 (MPX data service). However, message correlation is also required.

Two KMPs are defined: one for raw EDHOC and one for EDHOC over CoAP (RFC 7252), both making use of connection identifiers prepended to the messages, thereby supporting correlation also in the case of several EDHOC sessions in paralell.

1. Raw EDHOC over KMP.
2. The KMP payload transporting EDHOC message\_1 will prepend the CBOR simple type "true": (CBOR true, EDHOC message\_1).
3. EDHOC message 2 will prepend first the Initiator’s connection identifier (C\_I) resulting in (C\_I, EDHOC message 2).
4. The message transporting EDHOC message 3 will prepend C\_R in the format (C\_R, EDHOC message 3).
5. In case EDHOC message 4 is required, the message will prepend the C\_I to EDHOC message 4 resulting in the format (C\_I, EDHOC message 4).
6. EDHOC over CoAP (optionally using OSCORE as communication security protocol)

The prepended data carried over CoAP is described in Appendix A.1 of RFC 9528:

1. (CBOR true, EDHOC message\_1)
2. (EDHOC message\_2)
3. (C\_R, message\_3)
4. (message\_4)

When EDHOC message 4 is omitted, the KMP-FINISHED indication will be issued at the originator after sending EDHOC message 3, and at the recipient after successfully processing EDHOC message 3. In case EDHOC message\_4 is used, the KMP-FINISHED indication will be issued at the recipient after sending EDHOC message 4, and at the originator after receiving EDHOC message 4.

Ea.3.3 Algorithm negotiation

EDHOC defines cipher suites mapping as an ordered set of standardized algorithms to be used in the handshake and in a subsequent application protocol. EDHOC supports a very compact selection and negotiation of cipher suite protected from bidding down through a multisession procedure.

An EDHOC cipher suite consists of the following parameters:

* EDHOC AEAD algorithm
* EDHOC hash algorithm
* EDHOC MAC length in bytes (Static DH)
* EDHOC key exchange algorithm (ECDH curve)
* EDHOC signature algorithm
* application AEAD algorithm, and
* application hash algorithm.

Example: Ciphersuite number 3 is (AES-CCM-16-64-128, SHA-256, 8, P-256, ES256, AES‑CCM‑16‑64‑128, SHA-256)

Currently registered ciphersuites are shown in the IANA registry for EDHOC Cipher Suites: [https://www.iana.org/assignments/edhoc/edhoc.xhtml#edhoc-cipher-suites]. additional ciphersuites can be registered, for example replacing SHA-256 or using ASCON. Private ciphersuites are also allowed as indicated in the IANA registry.

Ea.3.4 Key derivation

The cryptographic core of EDHOC is based on the theoretical SIGMA-I protocol through its MAC-then-Sign variant and complemented with the key schedule inspired by the Noise XX pattern. The result is a common shared secret used to derive keying material unique for specific application instances based on the performed EDHOC session by means of an Exporter function.

Derived link key = EDHOC\_Exporter(exporter\_label, context, derived\_key\_length) = EDHOC\_KDF(PRK\_exporter, exporter\_label, context, length), where

* IANA is requested to register an exporter\_label (CBOR unsigned integer) for IEEE 802.15.9 from the "EDHOC Exporter Labels" registry
  + Different labels for 802.15.4 with raw EDHOC vs 802.15.4 with EDHOC over CoAP
* context is a CBOR byte string defined by the KMP to make the derived key unique [may be set to the empty string h'' if not needed for disambiguation]
  + context = (secKeyIdMode, ? secKeyIndex, ? secKeySource)
  + typical secKeyIdMode = 0x00 and the others omitted
* derived\_key\_length is a CBOR unsigned integer defined by the key length of the encryption algorithm. For example AES-CCM has derived\_key\_length = 16.

Ea.3.5 Broadcast and multicast key distribution

EDHOC does not distribute any broadcast or multicast keys. However, it enables the use of OSCORE that, in turn, allow group communication and the distribution of group keying material. Not in scope of this version of the standard.

References

[Bruni18]

Bruni, A., Sahl Jørgensen, T., Grønbech Petersen, T., and C. Schürmann, "Formal Verification of Ephemeral Diffie-Hellman Over COSE (EDHOC)", November 2018, <<https://www.springerprofessional.de/en/formal-verification-of-ephemeral-diffie-hellman-over-cose-edhoc/16284348>>.

[CottierPointcheval22]

Cottier, B. and D. Pointcheval, "Security Analysis of the EDHOC protocol", September 2022, <<https://arxiv.org/abs/2209.03599>>.

[Ferreira24]

L. Ferreira, "Computational Security Analysis of the Full EDHOC Protocol", May 2024,

<https://link.springer.com/chapter/10.1007/978-3-031-58868-6\_2>

[GuentherIlunga22]

Günther, F. and M. Mukendi, "Careful with MAc-then-SIGn: A Computational Analysis of the EDHOC Lightweight Authenticated Key Exchange Protocol", December 2022, <<https://eprint.iacr.org/2022/1705>>.

[Jacomme23]

Jacomme, C., Klein, E., Kremer, S., and M. Racouchot, "A comprehensive, formal and automated analysis of the EDHOC protocol", October 2022, <<https://hal.inria.fr/hal-03810102/>>.

[Norrman20]

Norrman, K., Sundararajan, V., and A. Bruni, "Formal Analysis of EDHOC Key Establishment for Constrained IoT Devices", September 2020, <<https://arxiv.org/abs/2007.11427>>.