Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: 15 September 2026

M. Chen
L. Su
China Mobile

14 March 2026

EDHOC Authenticated with AKA
draft-chen-lake-edhoc-aka-02

Abstract

This document defines an Authentication and Key Agreement (AKA) authentication method for the Ephemeral Diffie-Hellman Over COSE (EDHOC) key exchange protocol. This method, named EDHOC-AKA, is designed as a specific profile of the EDHOC Pre-Shared Key (PSK) authentication method defined in [draft-ietf-lake-edhoc-psk-06].

EDHOC-AKA leverages the 3GPP AKA protocol to dynamically generate a session-specific Pre-Shared Key, which is then used to run the EDHOC-PSK protocol flow. The AKA-specific parameters are transported within the External Authorization Data (EAD) fields of the EDHOC messages. This approach provides efficient mobile communication network access authentication for resource-constrained scenarios (such as Non-Terrestrial Networks, NTN) while inheriting the security properties of EDHOC-PSK, including mutual authentication, identity protection, and forward secrecy.

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 15 September 2026.

Copyright Notice

Table of Contents

1. Introduction 2
2. Terminology 3
3. Protocol 4
3.1. Relationship to EDHOC-PSK 4
3.2. Deriving the Session-Special PSK(K_AKA) 4
3.3. Message Flow of EDHOC-AKA 4
4. Key Derivation 5
5. Message Formating and Processing 5
5.1. Message1 6
5.2. Message2 6
5.3. Message3 6
5.4. Message4 6
6. IANA Considerations 7
6.1. EDHOC Method Type Registry 7
6.2. EDHOC EAD Label Registry 7
7. Security Considerations 7
8. Informative References 8
Authors' Addresses 9

1. Introduction

The Authentication and Key Agreement (AKA) protocol is a widely used mechanism for authenticating devices in mobile networks, as specified for 3G, 4G, and 5G systems RFC4187 [RFC5448] [RFC9048]. It relies on a long-term symmetric key pre-shared between the user's secure element (e.g., a SIM card) and the home network.

This document defines EDHOC-AKA, an authentication method for the Ephemeral Diffie-Hellman Over COSE (EDHOC) protocol [RFC9528] that utilizes the AKA mechanism. The primary innovation of EDHOC-AKA is that it does not define a new standalone protocol flow. Instead, it profiles the EDHOC Pre-Shared Key (PSK) authentication method specified in [draft-ietf-lake-edhoc-psk-06].

The core concept is as follows:

* The Initiator (e.g., a mobile device) and Responder (e.g., a service network) execute an AKA challenge-response exchange.

* This exchange is carried within the External Authorization Data (EAD) fields of the standard EDHOC message flow.

* Upon successful completion of the AKA exchange, both parties derive a shared, session-specific key, denoted here as K_AKA.

* This K_AKA is then used as the PSK required by the EDHOC-PSK authentication method.

This approach allows EDHOC-AKA to inherit the computational efficiency and security properties of EDHOC-PSK, such as identity protection against passive attackers and Perfect Forward Secrecy (PFS) against certain adversaries, while integrating seamlessly with established mobile network authentication infrastructure. This document specifies the mapping of AKA parameters to EAD fields, the derivation of K_AKA, and the necessary processing steps, while referencing [draft-ietf-lake-edhoc-psk-06] for the underlying protocol mechanics.

2. 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].

Readers are expected to be familiar with the terms and concepts described in EDHOC [RFC9528] and EDHOC-PSK [draft-ietf-lake-edhoc-psk-06].

* AKA: Authentication and Key Agreement. A challenge-response protocol based on a symmetric key.

* K: The long-term secret key shared between a user’s secure element and their home network.

* SUPI/SUCI: Subscription Permanent Identifier / Subscription Concealed Identifier. The user’s permanent and concealed identities in 5G.

* AKA Vector: A set of parameters generated by the network for an AKA challenge, typically including RAND (random challenge), AUTN (authentication token), XRES (expected response), CK (ciphering key), and IK (integrity key).

* K_AKA: A session-specific key derived from the AKA-generated keys (CK, IK), which serves as the PSK for the EDHOC-PSK protocol.

3. Protocol

3.1. Relationship to EDHOC-PSK

EDHOC-AKA is a profile of EDHOC-PSK and follow the protocol flow, key derivation logic, and message formatting specified in [draft-ietf-lake-edhoc-psk-06]. The fundamental difference lies in how the PSK is obtained. In a typical EDHOC-PSK scenario, the PSK is a static key provisioned out-of-band. In EDHOC-AKA, the PSK is dynamically derived in each session via an AKA exchange.

This AKA exchange is performed using the EAD mechanism defined in [RFC9528].

3.2. Deriving the Session-Special PSK(K_AKA)

The Responder (network) initiates an AKA challenge by obtaining an AKA vector from the home network based on the Initiator’s identity. The Initiator responds to the challenge. If the exchange is successful, both parties derive the session keys CK and IK.

These keys are then used to derive K_AKA, which will be used as the PSK for the current EDHOC session. The key derivation function (KDF) for this purpose is application-specific but MUST produce a key of the length required by the selected EDHOC cipher suite. For example, a common method is to use the EDHOC-KDF:

K_AKA = EDHOC-KDF(CK, "K_AKA", IK, desired_key_length)

The context string "K_AKA" ensures that the derived key is unique to this purpose.

3.3. Message Flow of EDHOC-AKA

The message flow is identical to that of EDHOC-PSK, as shown in Figure 1 of [draft-ietf-lake-edhoc-psk-06]. The AKA-specific parameters are transported in EAD fields as follows:

* EAD_1: Sent by the Initiator in message_1. It contains the Initiator’s identity (e.g., SUCI) needed by the Responder to fetch the correct AKA vector.

* EAD_2: Sent by the Responder in message_2. It contains the AKA challenge parameters (RAND and AUTN).

* EAD_3: Sent by the Initiator in message_3. It contains the AKA response (RES).

* EAD_4: Sent by the Responder in message_4. It MAY contain a final confirmation of the AKA authentication result from the network.

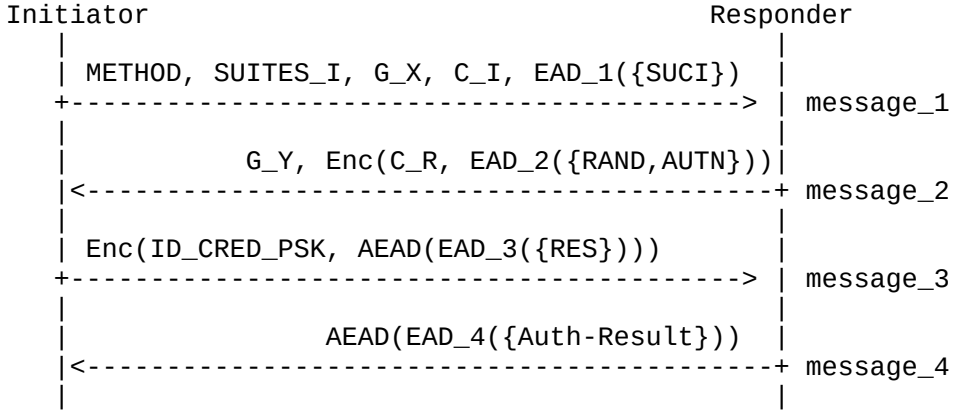

Figure 1: Overview of Message Flow of EDHOC-AKA

The field ID_CRED_PSK is used as defined in [draft-ietf-lake-edhoc-psk-06]. It may contain an identifier related to the ongoing AKA session or the long-term subscription to help the Responder manage keying material.

4. Key Derivation

The key derivation schedule for EDHOC-AKA SHALL be identical to the one specified in Section 4 of [draft-ietf-lake-edhoc-psk-06].

The key derivation for PRK_4e3m is:

PRK_4e3m = EDHOC_Extract(SALT_4e3m, PSK)

For EDHOC-AKA, the PSK used in this formula is the session-specific K_AKA derived from the AKA exchange, as described in Section 3.2 of this document. All other PRKs and derived keys (e.g., PRK_out, K_3, IV_3) are computed exactly as specified in the EDHOC-PSK draft.

5. Message Formating and Processing

Message formatting and processing SHALL follow the specifications in Section 5 of [draft-ietf-lake-edhoc-psk-06]. This section outlines the additional processing steps related to the EAD fields for the AKA exchange.

5.1. Message1

Message 1 (Initiator -> Responder) Initiator: The Initiator composes message_1 as specified in [draft-ietf-lake-edhoc-psk-06]. The METHOD selected is EDHOC-AKA (TBD1). The Initiator MUST include an EAD_1 field containing its identity (e.g., SUCI), formatted with the appropriate EAD label (see Section 6.2). Responder: The Responder processes message_1 and uses the identity from EAD_1 to request an AKA vector from the home network.

5.2. Message2

Message 2 (Responder -> Initiator) Responder: After obtaining the AKA vector, the Responder composes message_2 as specified in [draft-ietf-lake-edhoc-psk-06]. It MUST include an EAD_2 field containing the RAND and AUTN values, each formatted with the appropriate EAD label. Initiator: The Initiator decrypts message_2 to retrieve EAD_2. It passes the AUTN and RAND to its secure element to verify the network’s authenticity and compute the response RES, as well as the keys CK and IK. If verification succeeds, it derives K_AKA as per Section 3.2.

5.3. Message3

Message 3 (Initiator -> Responder) Initiator: The Initiator composes message_3 as specified in Section 5.3.2 of [draft-ietf-lake-edhoc-psk-06]. It uses the derived K_AKA as the PSK for this process. It MUST include an EAD_3 field containing the computed RES, formatted with the appropriate EAD label. Responder: The Responder processes message_3 as specified in Section 5.3.3 of [draft-ietf-lake-edhoc-psk-06]. To do this, it must first derive the same K_AKA from the CK and IK it holds. A successful decryption of message_3 implicitly authenticates the Initiator. The Responder then extracts RES from EAD_3 and compares it with its expected XRES. If they match, the AKA authentication is fully successful.

5.4. Message4

Message 4 (Responder -> Initiator) As noted in [draft-ietf-lake-edhoc-psk-06], message_4 is required for mutual authentication. It proves to the Initiator that the Responder also successfully derived K_AKA and completed the protocol. The Responder MAY include an EAD_4 field to convey the final status of the authentication from the network’s perspective.

6. IANA Considerations

6.1. EDHOC Method Type Registry

IANA is requested to register the following entry in the "EDHOC Method Type" registry under the group name "Ephemeral Diffie-Hellman Over COSE (EDHOC)".

Value	Initiator Authentication Key	Responder Authentication Key
TBD1	EDHOC-AKA	EDHOC-AKA

Table 1: Addition to the EDHOC Method Type Registry.

NOTE: Suggested value: TBD1 = 5. RFC Editor: Remove this note.

6.2. EDHOC EAD Label Registry

IANA is requested to register the following entries in the "EDHOC EAD Label" registry.

Label	Description	Reference
TBD2	AKA SUCI	this document
TBD3	AKA RAND	this document
TBD4	AKA AUTN	this document
TBD5	AKA RES	this document
TBD6	AKA Auth-Result	this document

Figure 3: EAD labels

NOTE: Suggested values: TBD2-TBD6 = sequential integers. RFC Editor: Remove this note.

7. Security Considerations

As EDHOC-AKA is a profile of EDHOC-PSK, it inherits the security properties analyzed in Section 9 of [draft-ietf-lake-edhoc-psk-06]. This section discusses considerations specific to the AKA integration.

Mutual Authentication: Strong mutual authentication is achieved. The Initiator authenticates the network by validating AUTN in message_2. The network authenticates the Initiator through two mechanisms: first, by the successful decryption of message_3 (which requires the correct K_AKA), and second, by the explicit validation of RES from EAD_3. Identity Protection: The Initiator's permanent identity (SUPI) is protected by using a concealed identity (SUCI) in EAD_1, which is only sent in the first message. The rest of the exchange is protected by keys derived from the ephemeral Diffie-Hellman exchange, providing identity protection against passive attackers as per EDHOC-PSK. Forward Secrecy: The protocol provides PFS. Even if the long-term key K is compromised, past session keys remain secure because they are derived from the ephemeral DH shared secret G_XY. An attacker who compromises K can impersonate the user in future sessions but cannot decrypt past traffic. Key Binding: The final session keys (derived from PRK_out) are cryptographically bound to both the ephemeral DH secret (G_XY) and the AKA-derived session key (K_AKA). This provides strong resistance against key-compromise and misbinding attacks. AKA Vector Handling: The AKA vector components (RAND, AUTN) are transported in an encrypted portion of message_2. This protects them from passive observation, though not from an active attacker who can manipulate the first two messages. The security of the AKA mechanism itself relies on the integrity of AUTN, which is handled by the Initiator's secure element.

8. Informative References

[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>.

[RFC4187] Arkko, J. and H. Haverinen, "Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA)", RFC 4187, DOI 10.17487/RFC4187, January 2006, <https://www.rfc-editor.org/rfc/rfc4187>.

[RFC5448] Arkko, J., Lehtovirta, V., and P. Eronen, "Improved Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA')", RFC 5448, DOI 10.17487/RFC5448, May 2009, <https://www.rfc-editor.org/rfc/rfc5448>.

[RFC9048] Arkko, J., Lehtovirta, V., Torvinen, V., and P. Eronen, "Improved Extensible Authentication Protocol Method for 3GPP Mobile Network Authentication and Key Agreement (EAP-AKA')", RFC 9048, DOI 10.17487/RFC9048, October 2021, <https://www.rfc-editor.org/rfc/rfc9048>.

Authors' Addresses

Meiling Chen
China Mobile
BeiJing
China
Email: chenmeiling@chinamobile.com

Li Su
China Mobile
BeiJing
China
Email: suli@chinamobile.com