Post-Quantum EDHOC
Initiator and Responder using signature and/or KEM

Clément Papon & Crisitna Onete

Xlim - Limoges

March 16, 2026
IETF 125 Shenzhen
draft-papon-lake-pq-edhoc-00

Outline

1. Context and objectives
2. Post-Quantum EDHOC when Initiator knows Responder
3. Post-Quantum EDHOC - Authentication with Signature and/or KEM
4. Security considerations
5. Bytes analysis

1. Context and objectives
2. Post-Quantum EDHOC when Initiator knows Responder
3. Post-Quantum EDHOC - Authentication with Signature and/or KEM
4. Security considerations
5. Bytes analysis

Context and objectives

Context

- Post-quantum secure extension for EDHOC
- Use of post-quantum signatures (same workflow as classical signatures)
- Use of a KEM (different mechanism than Diffie–Hellman)
- Maintain the Identity Protection security property

Objectives

- Propose a variant of [draft-pocero-authkem-ikr-edhoc-01] where the Initiator authenticates using a PQ-signature
- Propose a post-quantum alternative to EDHOC methods 1–3 (trade-off between number of messages, message size, and computational cost)
- Discussions on security properties (identity protection)

1 Context and objectives

2 Post-Quantum EDHOC when Initiator knows Responder

3 Post-Quantum EDHOC - Authentication with Signature and/or KEM

4 Security considerations

5 Bytes analysis

Initiator authenticates with a PQ-signature

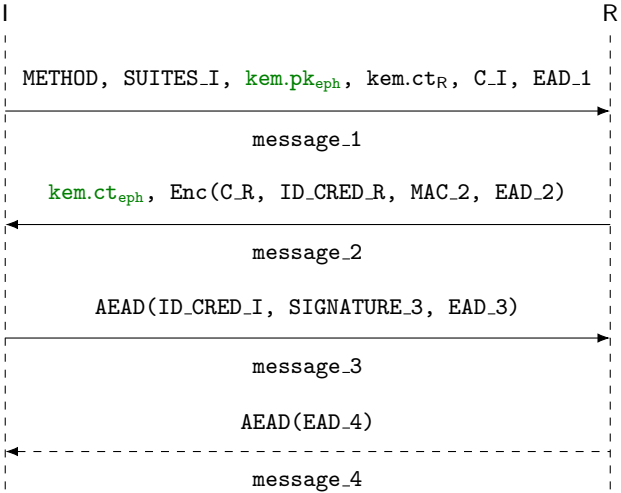

Figure: PQ-EDHOC-IKR, I Signs, R KEM message flow

Initiator authenticates with a PQ-signature

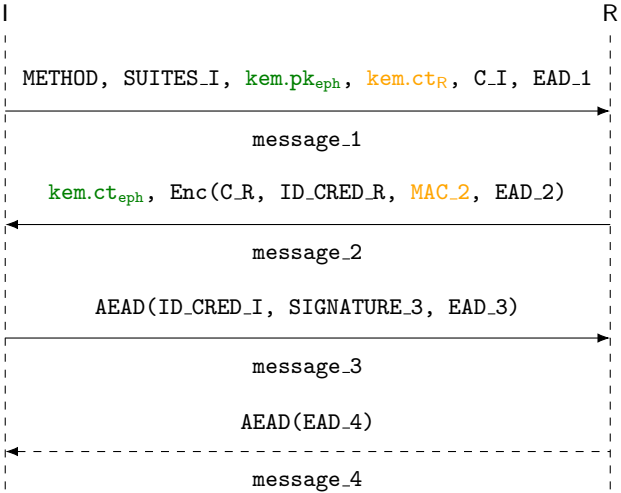

Figure: PQ-EDHOC-IKR, I Signs, R KEM message flow

Initiator authenticates with a PQ-signature

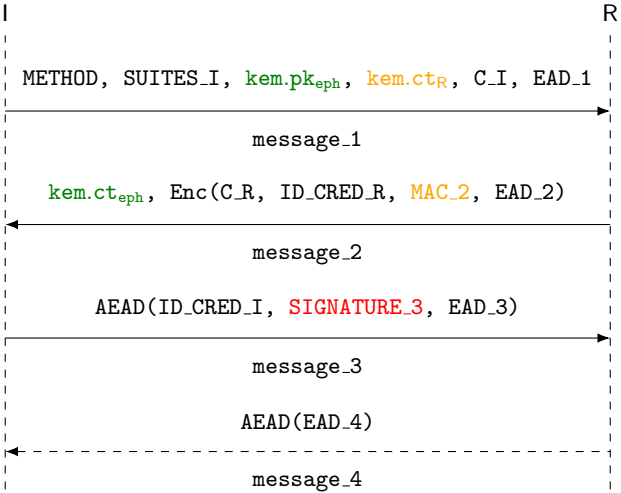

Figure: PQ-EDHOC-IKR, I Signs, R KEM message flow

1 Context and objectives

2 Post-Quantum EDHOC when Initiator knows Responder

3 Post-Quantum EDHOC - Authentication with Signature and/or KEM

4 Security considerations

5 Bytes analysis

Case 1: Initiator Signs, Responder KEM & Signs

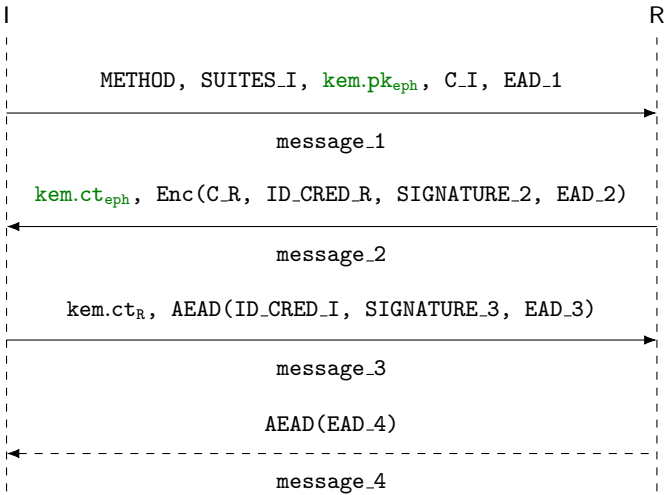

Figure: PQ-EDHOC, I Signs, R KEM & Signs message flow

Case 1: Initiator Signs, Responder KEM & Signs

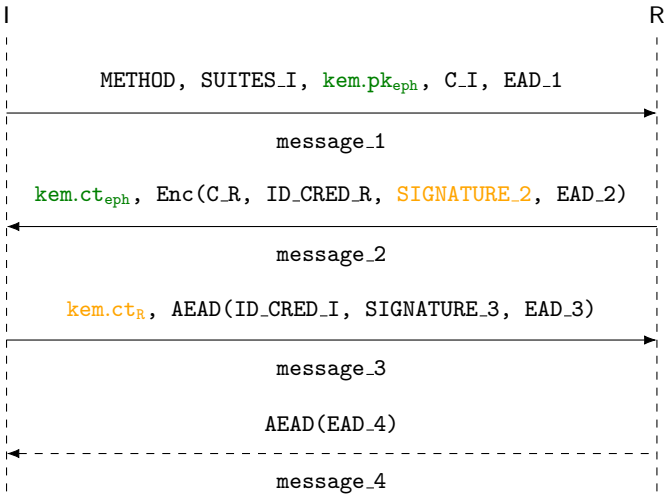

Figure: PQ-EDHOC, I Signs, R KEM & Signs message flow

Case 1: Initiator Signs, Responder KEM & Signs

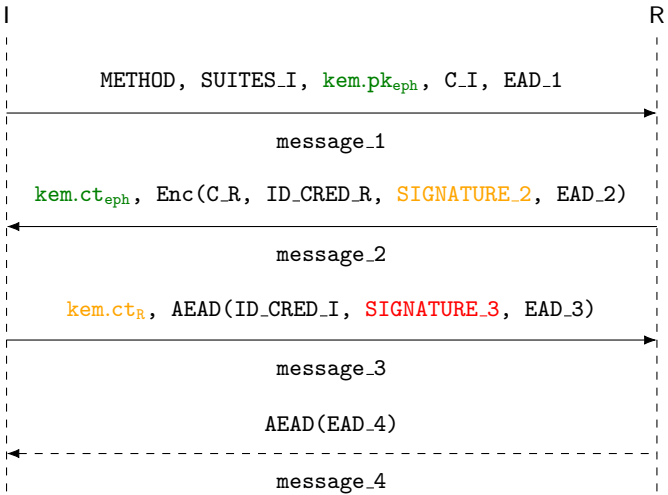

Figure: PQ-EDHOC, I Signs, R KEM & Signs message flow

Case 2: Initiator KEM & Signs, Responder KEM

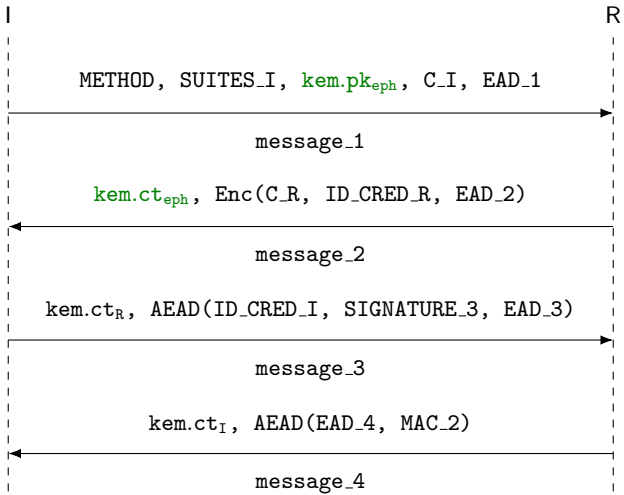

Figure: PQ-EDHOC, I KEM & Signs, R KEM message flow

Case 2: Initiator KEM & Signs, Responder KEM

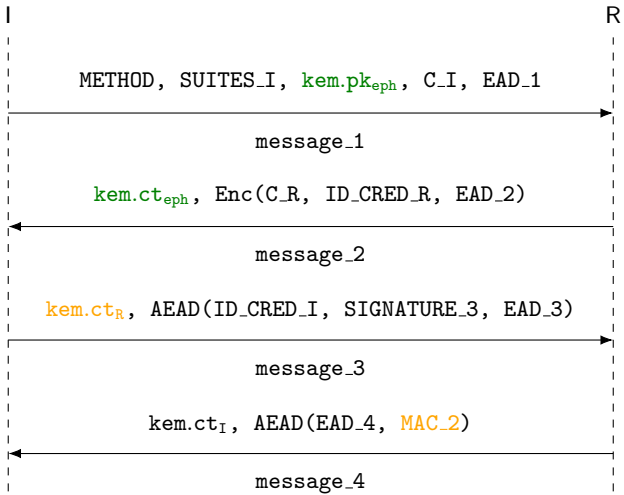

Figure: PQ-EDHOC, I KEM & Signs, R KEM message flow

Case 2: Initiator KEM & Signs, Responder KEM

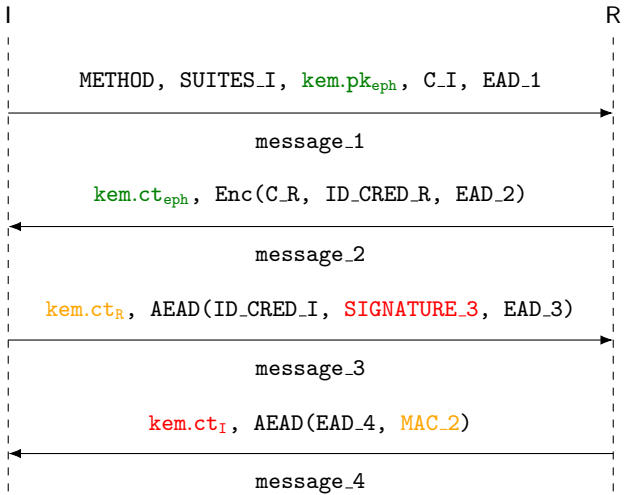

Figure: PQ-EDHOC, I KEM & Signs, R KEM message flow

1 Context and objectives

2 Post-Quantum EDHOC when Initiator knows Responder

3 Post-Quantum EDHOC - Authentication with Signature and/or KEM

4 Security considerations

5 Bytes analysis

Security considerations

- Global security: at the end of each handshake, both endpoints:
  - → securely compute the session key PRK_out
  - → securely authenticate their partner
- Forward secrecy
- Identity protection:
  - → each protocol verifies the Identity Protection security property
  - → we suggest a more global approach for this property, based on PPAKE (Privacy-Preserving Authenticated Key Exchange)
- Downgrade protection, Transcript hash binding, EAD security

1 Context and objectives

2 Post-Quantum EDHOC when Initiator knows Responder

3 Post-Quantum EDHOC - Authentication with Signature and/or KEM

4 Security considerations

5 Bytes analysis

Bytes comparison

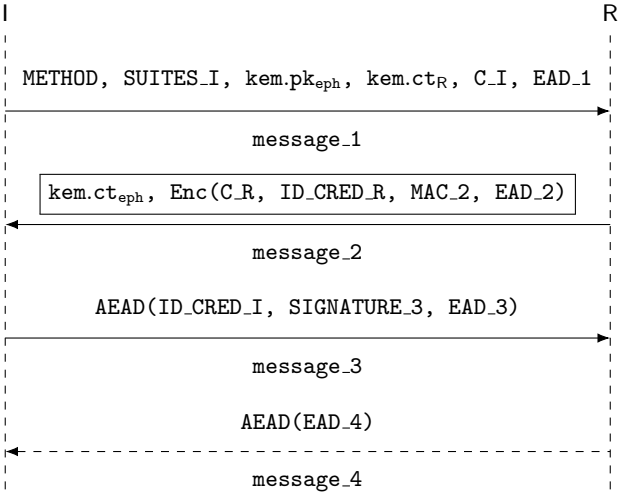

Figure: PQ-EDHOC-IKR, I Signs, R KEM message flow

Bytes comparison

$$\texttt{message\_2} = \underset{\downarrow}{\texttt{kem.ct}_{\texttt{eph}}} + \underset{\downarrow}{\texttt{Enc(C\_R,}} \; \underset{\downarrow}{\texttt{ID\_CRED\_R}} \; \underset{\downarrow}{\texttt{MAC\_2,}} \; \underset{\downarrow}{\texttt{EAD\_2)}}$$

$$\underset{768\ \text{bytes}}{} \qquad \underset{1\ \text{byte}}{} \quad \underset{4\ \text{bytes}}{} \quad \underset{8\ \text{bytes}}{} \quad \underset{0\ \text{byte}}{}$$

$$\text{Total} = 781\ \text{bytes}$$

- Connection Identifier, Method, Suites $\rightarrow$ 1 byte each
- Credentials (kid) $\rightarrow$ 4 or 14 bytes depending on authentication method
- AEAD: AES-CCM-16-64-128 $\rightarrow$ tag = 8 bytes
- MAC $\rightarrow$ 8 bytes
- KEM: ML-KEM-512 $\rightarrow$ pk = 800 bytes ; ciphertext = 768 bytes (FIPS 203)
- DSA: ML-DSA-44 $\rightarrow$ signature = 2420 bytes ; pk = 1312 bytes (FIPS 204)

Bytes comparison

	Protocol 1	Protocol 2	Protocol 3	Protocol 4	Protocol 5
message_1	1571	803	803	803	803
message_2	781	3203	3203	783	3203
message_3	2442	3210	2442	3210	3202
message_4	0	0	776	784	776
Total	4794	7216	7224	5580	7984

Figure: Bytes analysis of the 5 protocols.

Thank you!
Questions?