PQuAKE

Uri Blumenthal, Brandon Luo

July 2025

PQuAKE

Post-Quantum Authenticated Key Exchange

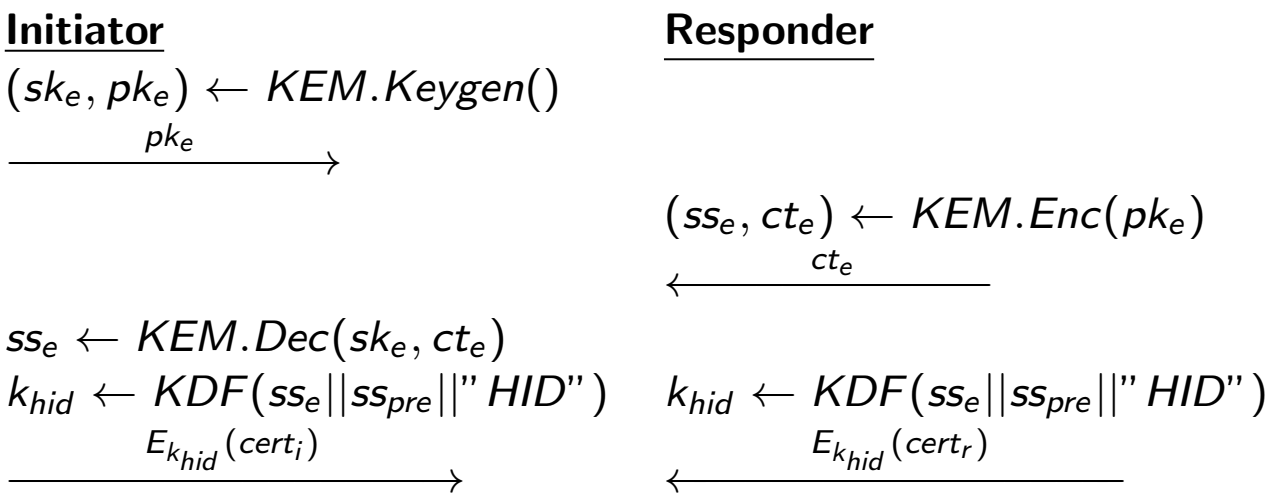
- Purpose: Minimize communication complexity for a post-quantum key exchange
- Properties: Compactness, Forward Secrecy, Mutual Authentication, Identity Hiding, Post-Quantum Security
- Expected Applications: Bandwidth- and power-constrained environments

Advantages

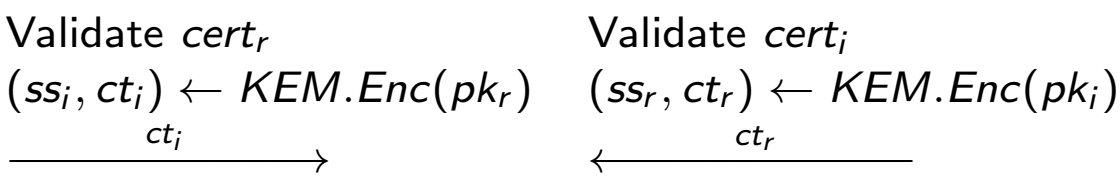
- Avoids online signature generation
- Exchanges fewer bytes
- Machine-checked security properties

PQuAKE (Part 1)

Establish confidential link and exchange certificates



Encapsulate and send shared secrets



PQuAKE (Part 2)

Hash the transcript

$$h^i_{sent}, h^r_{recv} \leftarrow Sha384(Sha384(Sha384(pk_e)||E_{k_{hid}}(cert_i))||ct_i)$$
$$h^r_{sent}, h^i_{recv} \leftarrow Sha384(Sha384(Sha384(ct_e)||E_{k_{hid}}(cert_r))||ct_r)$$

Decapsulate shared secrets and derive session keys

$$ss_r \leftarrow KEM.Dec(sk_r, ct_r)$$
$$S \leftarrow ss_e||ss_i||ss_r||ss_{pre}||h^i_{sent}||h^i_{recv}$$
$$k_{C_i}, k_{C_r}, k_{session} \leftarrow KDF(S)$$

$$ss_i \leftarrow KEM.Dec(sk_i, ct_i)$$
$$S \leftarrow ss_e||ss_i||ss_r||ss_{pre}||h^r_{recv}||h^r_{sent}$$
$$k_{C_i}, k_{C_r}, k_{session} \leftarrow KDF(S)$$

Key confirmation

$$C_i \leftarrow HMAC(k_{C_i}, h^i_{sent}||h^i_{recv})$$
$$\xrightarrow{C_i}$$
$$C_r =^? HMAC(k_{C_r}, h^i_{recv}||h^i_{sent})$$

$$C_r \leftarrow HMAC(k_{C_r}, h^r_{sent}||h^r_{recv})$$
$$\xleftarrow{C_r}$$
$$C_i =^? HMAC(k_{C_i}, h^r_{recv}||h^r_{sent})$$

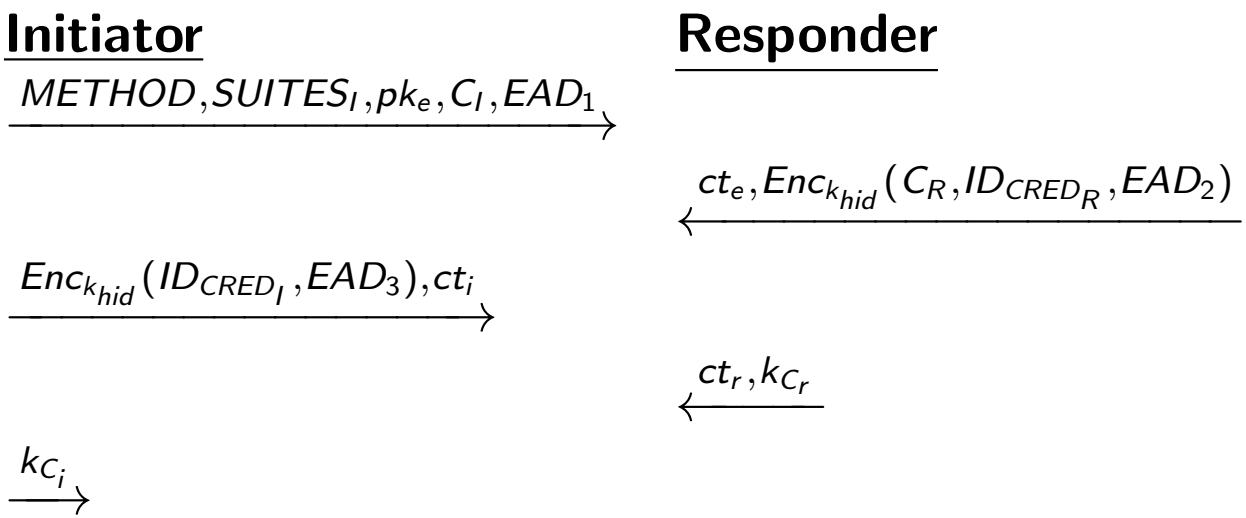
EDHOC Extension

Add new authentication method (implicit authentication)

- Method Type Value 4

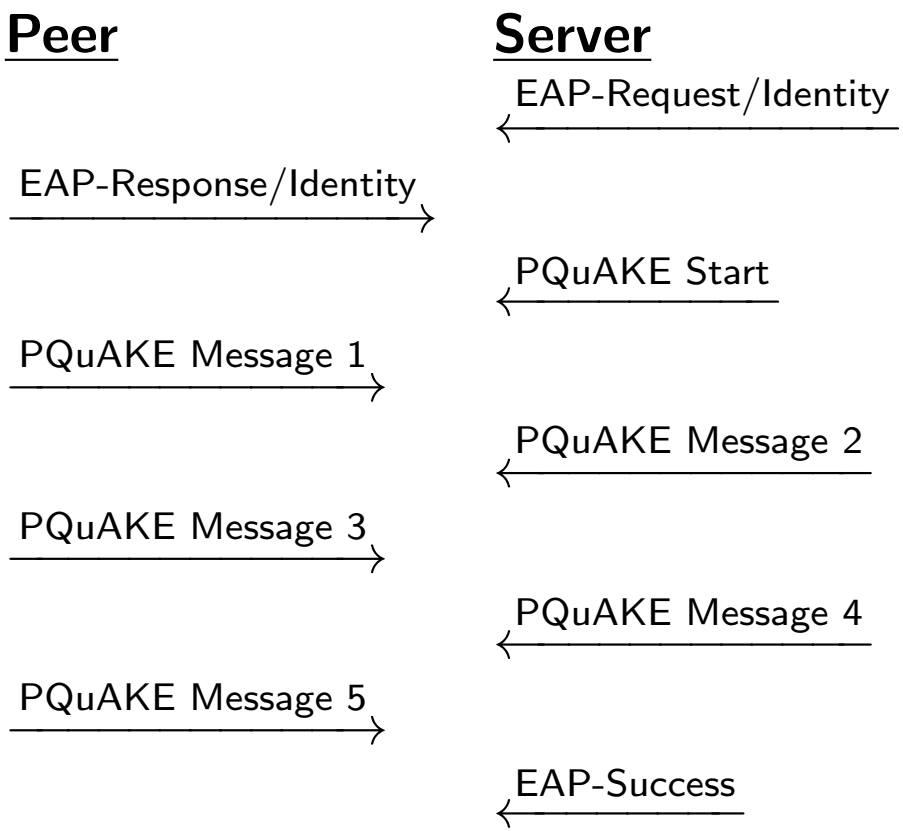
Add new cipher suites

- ML-KEM-512, ML-KEM-768, ML-KEM-1024



EAP-PQUAKE

New EAP method proposal: (EAP-PQUAKE)



Alternatively, integrated as part of EAP-EDHOC.

Performance Analysis

Operation	Initiator	Responder
KEM.Keygen	1	0
KEM.Encap	1	2
KEM.Decap	2	1
DSA.Verify	1	1
AES256-GCM	2	2
SHA384 calls	12	12
SHA384 rounds	175	175

Table: PQuAKE Computational Complexity

Round	Message	Size
1	$pk_e, ct_e, E_{k_{hid}}(id_r)$	$\approx 800 + 768 + 32$ bytes
2	$E_{k_{hid}}(id_i), ct_i, ct_r, k_{C_r}$	$\approx 32 + 768 + 768 + 48$ bytes
3	k_{C_i}	48 bytes
Total		≈ 3264 bytes

Table: PQuAKE Communication Complexity (cached certificates)

Summary

We propose an extension to the protocol (LAKE/EAP/IPsec) which

- has verified security properties
- exchanges fewer bytes than traditional approaches
- avoids online signatures (better real-time properties and code-size)

Next Steps:

- Add optimizations like cached/preloaded certificates
- Collaborate with EDHOC/EAP/IKEv2 working groups to determine best method of integration
- Discuss at SAAG/CRFG the problem of minimizing communication for post-quantum protocols