

1 Hybrid Cryptosystems

1.1 Motivation and Basic Idea

A hybrid cryptosystem combines classical and post-quantum cryptographic mechanisms to reduce dependence on a single security assumption during the transition to quantum-resistant cryptography. For example, ECDH relies on the elliptic-curve discrete logarithm problem, which is vulnerable to Shor's algorithm, whereas ML-KEM is based on lattice-related assumptions believed to resist known quantum attacks. Hybrid security, however, depends on the construction and its security analysis.

1.2 Hybrid Key Establishment

Hybrid key establishment combines a classical mechanism, such as elliptic-curve Diffie–Hellman (ECDH), with a post-quantum key encapsulation mechanism (KEM), such as ML-KEM. The parties independently derive classical and post-quantum shared secrets:

$$K_C = \text{ECDH}(sk_A, pk_B), \quad K_P = \text{ML-KEM.Decap}(sk_B, c_P).$$

They combine these secrets using a suitable key-derivation function:

$$K_H = \text{KDF}(K_C \| K_P, \text{context}),$$

where the context binds the derived key to protocol and session information. The resulting key can protect application data using authenticated encryption, such as AES-GCM.

A properly designed combiner can preserve security if either component remains secure, but not if both are compromised. Secure transcript binding, failure handling, and protection against downgrade attacks are essential.

Example: TLS 1.3. A hybrid key-establishment mechanism can combine X25519 with ML-KEM-768 to derive handshake and application traffic secrets through the specified hybrid construction and TLS key schedule. Algorithm negotiation and exchanged values must be bound to the handshake transcript.

1.3 Hybrid Digital Signatures

A hybrid signature combines classical and post-quantum signatures over the same message. For example, Alice can use ECDSA and ML-DSA:

$$\sigma_C = \text{Sign}_C(sk_C, m), \quad \sigma_P = \text{Sign}_P(sk_P, m).$$

The hybrid signature is

$$\sigma_H = (\sigma_C, \sigma_P).$$

Under an *AND verification policy*, the verifier accepts only if both signatures are valid:

$$\text{Verify}_H(m, \sigma_H) = 1 \iff \text{Verify}_C(pk_C, m, \sigma_C) = 1 \wedge \text{Verify}_P(pk_P, m, \sigma_P) = 1.$$

Under suitable composition assumptions, this policy can preserve unforgeability if at least one component remains secure. An *OR verification policy*, which accepts either valid signature, may improve compatibility but does not provide the same protection against compromise of either component. Public-key identity binding, message consistency, and the verification policy must be specified explicitly.

In TLS 1.3, hybrid key establishment and hybrid signature authentication serve different purposes. Combining X25519 with ML-KEM provides hybrid key establishment, whereas combining ECDSA with ML-DSA provides hybrid authentication, subject to protocol and implementation support.

1.4 Practical Considerations

Hybrid deployment can facilitate migration to post-quantum security, but may increase computational costs, communication overhead, certificate sizes, and implementation complexity. In a public-key infrastructure (PKI), secure deployment may require changes to certificates, trust stores, and validation procedures. Using multiple algorithms or public keys alone does not guarantee hybrid security.

2 Post-Quantum Cryptography (PQC) Integration

PQC integration is the process of incorporating quantum-resistant cryptographic algorithms into existing applications, protocols, and security infrastructure. Unlike hybrid cryptography, which combines multiple mechanisms, PQC integration is a broader migration process that includes algorithm replacement, hybrid deployment, and protocol modification.

2.1 Approaches to PQC Integration

Direct Replacement. Classical algorithms can be replaced with post-quantum alternatives. For example, ECDH can be replaced by ML-KEM (NIST FIPS 203), while classical signature schemes can be replaced by ML-DSA (FIPS 204) or SLH-DSA (FIPS 205). This approach may require changes to protocols, libraries, certificates, and applications.

Hybrid Integration. Classical and post-quantum mechanisms are used together during migration. For example, ECDH and ML-KEM can establish independent shared secrets, which are combined using a suitable KDF. This approach reduces dependence on a single cryptographic assumption, provided that the combiner and surrounding protocol are secure.

Protocol and Infrastructure Integration. PQC deployment may require updates to TLS algorithm negotiation, key derivation, and transcript binding. PKI migration may also require changes to certificate formats, signature algorithms, trust stores, and certificate-validation procedures.

2.2 Challenges in PQC Integration

Key challenges include:

- **Performance and overhead:** Different computational costs and larger public keys, ciphertexts, or signatures.
- **Compatibility:** Limited support in legacy protocols, devices, libraries, and applications.
- **Security:** Secure key management, implementation, algorithm negotiation, and downgrade prevention.
- **Cryptographic agility:** Supporting controlled algorithm updates without redesigning the entire system.
- **Migration:** Identifying cryptographic dependencies and testing interoperability across legacy and updated components.

2.3 Example: Secure Communication Migration

A client-server application using ECDH for key establishment and classical signatures for authentication can migrate to PQC through direct replacement or hybrid deployment. Direct replacement uses ML-KEM and suitable post-quantum signatures, while hybrid deployment combines classical and post-quantum mechanisms where required. The final configuration should be selected after security analysis, interoperability testing, and policy evaluation, ensuring secure key derivation, authentication, negotiation, and failure handling.