Hybrid Certificate Chains and Post-Quantum Trust Anchors for Validator Networks

Zach Kelling

Lux Industries

2024

Abstract

We describe Lux Network's hybrid X.509 certificate infrastructure for validator identity, combining classical ECDSA/BLS signatures with post-quantum ML-DSA signatures in a single certificate chain. Each validator holds a dual-signature certificate: one classical signature for compatibility with existing TLS infrastructure, one ML-DSA signature for quantum resistance. Trust anchors are rooted in Q-Chain governance, with on-chain registration of certificate fingerprints and revocation via stake-weighted voting. We specify the certificate profile, the enrollment protocol, the dual-signature verification procedure, and the integration with device identity for hardware-attested validator nodes. The system is deployed on Lux mainnet, securing validator-to-validator TLS and staking key binding.

1 Introduction

Validator identity in proof-of-stake blockchains rests on public key infrastructure. In Lux, each validator is identified by a TLS certificate (staker.crt) that binds a NodeID to a public key. This certificate is used for peer-to-peer authentication, message signing, and stake registration. If the signature scheme underlying these certificates is broken, the entire validator set's identity collapses.

The NIST post-quantum standardization [1] provides new signature schemes (ML-DSA, SLH-DSA) that resist quantum attacks. However, replacing classical certificates wholesale introduces risk: the new schemes are less battle-tested, their implementations are younger, and a flaw in the PQ scheme during a pure-PQ deployment would be catastrophic.

Hybrid certificates address this by requiring *both* a classical and a post-quantum signature on every certificate. An attacker must break both schemes simultaneously to forge a validator identity. This paper specifies:

1. The hybrid certificate profile for Lux validators.
2. The dual-signature trust anchor rooted in Q-Chain.
3. The enrollment protocol for new validators.
4. Device identity integration for hardware-attested nodes.

2 Certificate Profile

2.1 Structure

A Lux hybrid certificate extends X.509v3 with a custom extension carrying the post-quantum signature and public key. The certificate is valid if and only if both signatures verify.

```
Certificate:
  Version: 3
  Serial: <unique>
  Issuer: CN=Lux Validator CA, O=Lux Network
  Subject: CN=NodeID-<base58>, O=Lux Network
  Public Key (classical): BLS12-381 (48 bytes)
  Extensions:
    x-lux-pq-pubkey: ML-DSA-65 (1952 bytes)
    x-lux-pq-signature: ML-DSA-65 sig (3309 bytes)
    x-lux-node-id: NodeID-<base58>
```

```
    x-lux-bls-pop: BLS proof-of-possession
  Signature Algorithm: BLS12-381 (classical)
  Signature: <BLS signature over TBS>
```

2.2 Dual-Signature Verification

1: **function** VERIFYHYBRIDCERT(cert)
2: $tbs \leftarrow cert.\text{toBeSigned}$
3: $valid_c \leftarrow \text{VerifyBLS}(cert.\text{classicalPubKey}, cert.\text{classicalSig}, tbs)$
4: $valid_{pq} \leftarrow \text{VerifyMLDSA}(cert.\text{pqPubKey}, cert.\text{pqSig}, tbs)$
5: **return** $valid_c \wedge valid_{pq}$
6: **end function**

Both signatures cover identical to-be-signed (TBS) data. The PQ signature is placed in an extension rather than replacing the primary signature to maintain compatibility with standard TLS libraries that parse the classical signature field.

2.3 Size Analysis

Component	Classical Only	Hybrid
Public key (classical)	48 B	48 B
Public key (PQ)	—	1,952 B
Signature (classical)	96 B	96 B
Signature (PQ)	—	3,309 B
Proof of possession	96 B	96 B
X.509 overhead	∼200 B	∼300 B
Total	∼440 B	∼5,801 B

Table 1: Certificate sizes. The hybrid overhead is ∼5.4 KB per validator, negligible for a set of 21–1000 validators.

3 Trust Anchors

3.1 On-Chain Root of Trust

The trust anchor for Lux validator certificates is not a traditional CA hierarchy but the Q-Chain validator registry. When a validator registers its stake, it submits both the classical and PQ public keys. Q-Chain stores the binding:

$$\text{Registry}[\text{NodeID}] = (\text{pk}_{\text{BLS}}, \text{pk}_{\text{MLDSA}}, \text{stakeAmount}, \text{epoch}_{\text{registered}})$$

A certificate is trusted if and only if its public keys match the on-chain registration for the claimed NodeID. This eliminates the need for a separate certificate authority—the blockchain itself is the CA.

3.2 Revocation

Certificate revocation occurs through two mechanisms:

1. **Stake withdrawal**: Removing stake from Q-Chain automatically revokes the associated certificate. Peers reject connections from unstaked NodeIDs.

2. **Governance revocation**: A governance proposal can revoke a specific NodeID's certificate (for key compromise, misbehavior). This requires a stake-weighted $> 2/3$ vote.

No Certificate Revocation Lists (CRLs) or OCSP responders are needed. Peers query Q-Chain state directly—or use the local state copy synchronized via the consensus protocol.

4 Enrollment Protocol

New validator enrollment follows a four-step protocol:

1. **Key Generation**: The validator generates a BLS key pair and an ML-DSA-65 key pair locally. If using an HSM, the private keys are generated inside the secure boundary [3].

2. **Certificate Request**: The validator constructs a self-signed hybrid certificate containing both public keys and submits it to Q-Chain as part of the stake registration transaction.

3. **On-Chain Validation**: Q-Chain validates the BLS proof-of-possession, verifies the ML-DSA self-signature, and registers the public keys if the stake meets the minimum (2,000 LUX).

4. **Peer Distribution**: Once registered, the validator's certificate is distributed to all peers via the standard P2P gossip protocol. Peers verify the certificate against Q-Chain state before accepting connections.

function EnrollValidator(stake, blsKey, mldsaKey)
$cert \leftarrow$ SelfSignHybrid($blsKey, mldsaKey$)
$pop \leftarrow$ BLSProofOfPossession($blsKey$)
$tx \leftarrow$ StakeRegistration($stake, cert, pop$)
Submit(tx, Q-Chain)
end function

5 Device Identity

For hardware-attested validator nodes, the hybrid certificate chain extends to include a device identity layer:

```
Trust Chain:
  Q-Chain Registry (root of trust)
    -> Validator Hybrid Certificate (BLS + ML-DSA)
      -> Device Attestation Certificate (TPM/TEE)
        -> Runtime Measurement (PCR values / TEE quote)
```

A validator running in a Trusted Execution Environment (TEE) or with a TPM can bind its hybrid certificate to a hardware attestation [4]. This provides assurance that the private keys are held in tamper-resistant hardware and that the validator software has not been modified.

The device attestation is optional—validators without hardware attestation still participate, but their certificates lack the device identity extension. Delegators can use this distinction to prefer hardware-attested validators.

6 Security Analysis

Theorem 1 (Hybrid Binding)**.** *An adversary who compromises only the classical signature scheme cannot forge a valid hybrid certificate.*

Proof. A valid hybrid certificate requires both $\text{VerifyBLS}(\cdot) = \text{true}$ and $\text{VerifyMLDSA}(\cdot) = \text{true}$ over identical TBS data. Compromising BLS allows forging σ_{BLS} but not σ_{MLDSA} (which remains secure under lattice hardness). The forged certificate fails the dual-signature check. □

Theorem 2 (On-Chain Revocation Completeness)**.** *A revoked certificate is rejected by all honest peers within one consensus round after the revocation transaction is finalized.*

Proof. Honest peers verify certificates against their local copy of Q-Chain state. Q-Chain finality (via Quasar) occurs within 10–100 ms [2]. Once finalized, all honest peers observe the revocation within one state sync interval. Peers that have not yet synced reject the certificate upon next sync. □

7 Conclusion

Hybrid certificates provide defense-in-depth for validator identity: breaking either the classical or post-quantum scheme alone is insufficient to forge an identity. By rooting trust in Q-Chain rather than a traditional CA hierarchy, Lux eliminates external trust dependencies and integrates revocation into the consensus protocol. The enrollment protocol requires no manual CA operations, and device identity integration provides optional hardware attestation for delegators who demand it.

References

[1] NIST (2024). *Post-Quantum Cryptography Standardization: FIPS 203, 204, 205*. National Institute of Standards and Technology.

[2] Kelling, Z. (2025). *Quasar: Quantum-Secure Multi-Engine Consensus with Triple-Proof Quantum Finality*. Lux Industries Research.

[3] Kelling, Z. (2025). *Secure Enclave, HSM, and Threshold Boundary Design*. Lux Industries Research.

[4] Kelling, Z. (2024). *TEE Computing Mesh for Attestable Validator Infrastructure*. Lux Industries Research.

[5] Kelling, Z. (2024). *Post-Quantum Cryptographic Suite for Blockchain Infrastructure*. Lux Industries Research.

[6] Kelling, Z. (2024). *Cryptographic Agility Architecture for Long-Lived Blockchain Systems*. Lux Industries Research.

[7] Laurie, B. et al. (2021). *Certificate Transparency Version 2.0*. RFC 9162.