Triple-Proof Quantum Finality: Post-Quantum Consensus for the Lux Network

Zach Kelling

Lux Industries

April 2026

Triple-Proof Quantum Finality

Post-Quantum Consensus for the Lux Network

Lux Industries, Inc.

April 2026 — v1.0

Abstract

We present **triple-proof quantum finality**, a consensus mechanism for the Lux Network that achieves simultaneous classical speed, post-quantum security, and validator privacy in 248 bytes per epoch. The mechanism combines three independent cryptographic proofs: (1) BLS12-381 aggregate signatures for sub-millisecond classical consensus, (2) STARK-aggregated ML-DSA-65 (FIPS 204) signatures for post-quantum identity binding, and (3) Corona lattice-based threshold signatures for anonymous validator participation. No single cryptographic break compromises the system. At 1ms block times, the mechanism requires approximately 1 TB/year of storage for block headers and epoch proofs combined. We provide benchmarks showing ∼2–5 ms CPU QuasarCert verification on commodity hardware [7].

Introduction

Post-quantum cryptography faces a fundamental tension: the most promising post-quantum signature schemes (ML-DSA, SLH-DSA) produce signatures 50–100× larger than classical schemes (ECDSA, BLS), making them impractical for per-block consensus in high-throughput blockchains. Signature aggregation, which makes BLS practical for consensus (N signatures → 48 bytes), has no known post-quantum equivalent.

We resolve this tension with a layered approach:

1. **Per-block**: Use BLS aggregation for speed (48 bytes, sub-millisecond).
2. **Per-epoch**: Use STARK proofs to compress N ML-DSA signatures into ∼200 bytes.
3. **Embedded**: Corona threshold signatures provide privacy within the STARK circuit at zero additional cost.

The result is a consensus mechanism that is simultaneously fast (1ms blocks), compact (248 bytes/epoch), post-quantum secure (ML-DSA + Corona), and privacy-preserving (anonymous validator participation).

Threat Model

We consider three classes of adversary:

- **Classical adversary**: Can corrupt up to $f < N/3$ validators. Cannot break BLS, ML-DSA, or Corona.
- **Quantum adversary**: Has access to a cryptographically relevant quantum computer (CRQC). Can break BLS and ECDSA. Cannot break ML-DSA (NIST FIPS 204) or Corona (Module-LWE).

- **Future adversary**: One of ML-DSA or Corona has an unforeseen mathematical weakness. The other, combined with BLS, still provides security.

Triple-proof finality is secure against all three adversary classes simultaneously.

The Three Proofs

Proof 1: BLS Aggregate Signature (Classical Speed)

Every validator signs each block with their BLS12-381 key. The block proposer aggregates all signatures into a single 48-byte aggregate signature. Verification is $O(1)$ regardless of validator count.

Property	**Value**
Curve	BLS12-381
Security	128-bit classical
Signature size	96 bytes (individual), 48 bytes (aggregated)
Aggregation	$N \to 1$
Verification	$O(1)$ pairing check
Post-quantum	No

BLS provides the real-time consensus path. It is fast, compact, and well-understood. Its lack of post-quantum security is compensated by Proofs 2 and 3.

Proof 2: STARK-Aggregated ML-DSA (Post-Quantum Identity)

Every validator also signs each epoch checkpoint with their ML-DSA-65 key (FIPS 204, ML-DSA). Individual ML-DSA-65 signatures are 3,309 bytes each — too large to include in block headers.

The STARK aggregation step compresses N ML-DSA signatures into a single succinct proof of approximately 200 bytes:

$$\text{STARK}(\{(\text{pk}_i, \sigma_i, m)\}_{i=1}^{N}) \to \pi \quad |\pi| \approx 200 \text{ bytes}$$

The STARK proof attests: "N valid ML-DSA-65 signatures over message m exist, signed by public keys $\{\text{pk}_i\}$." Verification of π is $O(\log N)$.

Property	**Value**
Signature scheme	ML-DSA-65 (FIPS 204)
Security	128-bit post-quantum (NIST Level 3)
Individual signature	3,309 bytes
Public key	1,952 bytes
Aggregation	STARK proof ($\sim$200 bytes)
Verification	$O(\log N)$
QuasarCert verify	$\sim$2–5 ms CPU [7]

If BLS is broken by a quantum computer, the STARK proof of ML-DSA signatures remains valid. The entire chain history is verifiable post-quantum.

Proof 3: Corona Threshold Signature (Post-Quantum Privacy)

Corona [4] is a lattice-based (Module-LWE, Ringtail-derived) threshold signature scheme with an anonymity property: the verifier can confirm that a threshold of validators signed, but cannot determine *which* validators participated.

Corona operates in two rounds:

1. **Round 1**: Each participating validator generates a commitment share.
2. **Round 2**: Shares are combined into a threshold signature. The anonymity set is the full validator committee.

The Corona signature is embedded within the STARK circuit — the STARK proof simultaneously validates both ML-DSA identity signatures and Corona threshold signatures, adding zero bytes to the epoch proof.

Property	**Value**
Basis	Module-LWE (lattice)
Security	128-bit post-quantum
Threshold	t-of-N
Anonymity	Yes (ring signature property)
Rounds	2
Additional bytes	0 (embedded in STARK proof)

Corona provides two properties: (1) post-quantum security at a Module-LWE parameter set distinct from ML-DSA's, and (2) validator privacy — no external observer can determine which specific validators signed a given epoch.

Block Structure

Per-Block Header (every 1ms)

```
BlockHeader {
    parent_hash:    [32]byte    // Parent block hash
    state_root:     [32]byte    // Merkle root of state
    tx_root:        [32]byte    // Merkle root of transactions
    timestamp:      uint64      // 8 bytes
    height:         uint64      // 8 bytes
    bls_aggregate:  [32]byte    // Truncated BLS aggregate hash
    // Total: 144 bytes
}
```

The `bls_aggregate` field contains a 32-byte hash of the full 48-byte BLS aggregate signature. The full signature is recoverable from any validator's archive.

Per-Epoch Proof (every 1,000 blocks = 1 second)

```
EpochProof {
    bls_aggregate:  [48]byte    // Full BLS aggregate signature
    zk_proof:       [~200]byte  // STARK(ML-DSA sigs + Corona threshold)
    validators:     uint16      // Count of participating validators
    epoch:          uint64      // Epoch number
```

```
    // Total: ~258 bytes
}
```

Security Analysis

Defense in Depth

The triple-proof design rests on three independent mathematical assumptions:

1. **Discrete logarithm** (BLS): Hardness of the elliptic curve discrete logarithm problem on BLS12-381.
2. **Lattice problems** (ML-DSA + Corona): Hardness of Module-LWE (both legs are Module-LWE, at distinct parameter sets).
3. **Hash functions** (STARK): Collision resistance of the hash function used in the STARK proof system.

Any two of these three assumptions suffice for full security:

Scenario	BLS	ML-DSA/RT	STARK
Classical adversary	✓	✓	✓
Quantum breaks DL	×	✓	✓
Lattice weakness found	✓	×	✓
STARK hash broken	✓	✓(individual sigs)	×
Quantum + lattice	×	×	✓*

*If both discrete log and lattice problems are broken, only the STARK proof (hash-based) remains. This provides integrity but not authenticity — a scenario considered extremely unlikely given current understanding of quantum computing and lattice cryptography.

Quantum Attack Window

A quantum adversary attempting to forge a BLS signature must do so within the epoch window (1 second). Current estimates for breaking BLS12-381 with a CRQC require $> 10^6$ logical qubits operating for minutes to hours. The 1-second epoch window is physically impossible to exploit with foreseeable quantum technology.

Even if the epoch window were exploitable, the ML-DSA + Corona proofs would detect the forgery at the next epoch boundary.

Performance

Benchmarks

Measured on Apple M1 Max (10 CPU cores, darwin/arm64, Go 1.26.1). GPU numbers are estimated; all others are `go test -bench` output [7].

Operation	Time
BLS sign	350 μs
BLS verify (single)	820 μs
BLS aggregated verify (N=100)	875 μs
ML-DSA-65 sign	495 μs
ML-DSA-65 verify (raw)	181 μs
ML-DSA-65 verify (via fx)	254 μs
ML-DSA-65 verify (cached)	3 μs
Groth16 prover (CPU)	∼400 ms
Groth16 prover (GPU, est.)	∼5–15 ms
Groth16 verify (3 pairings, CPU)	∼1–3 ms
Quasar block processing (BLS+ML-DSA+Corona)	1.85 ms
QuasarCert total CPU verify	**∼2–5 ms**

Storage

At 1ms block times (1,000 blocks/second):

Component	Size	Frequency	Annual
Block header	144 B	1,000/s	4.5 TB
Epoch proof	258 B	1/s	8.1 GB
Total (without tx data)			**∼4.5 TB**

With pruning (retain only epoch proofs after 30 days):

Component	Retained	Annual
Recent blocks (30 days)	144 B × 1000/s × 30d	373 GB
Epoch proofs (permanent)	258 B × 1/s × 365d	8.1 GB
Total (pruned)		**∼381 GB**

Application Chains

Any chain running on the Lux primary network inherits triple-proof quantum finality. The Lux validator set provides consensus for all chains simultaneously.

Current application chains:

Chain	Operator	Purpose	VM
Application Chain A	Partner	Regulated securities (ATS)	EVM + precompiles
Application Chain B	Partner	On-chain orderbook	Custom CLOB
Application Chain C	Partner	Encrypted computation	Threshold FHE
Application Chain D	Partner	AI/DeSci research	Custom

All chains benefit from the same 248-byte quantum epoch proofs without any chain-specific implementation. Quantum finality is a property of the Lux consensus layer, not the application layer.

Validator Key Set

Each Lux validator maintains four cryptographic keys:

Key Type	Crypto	Purpose	NIST Standard
EC	secp256k1 (ECDSA)	EVM transactions, addresses	—
BLS	BLS12-381	Consensus aggregation	—
ML-DSA	Dilithium-65	PQ identity	FIPS 204
Corona	Module-LWE	PQ anonymous signing	—

The ML-DSA key is the validator's *long-term identity*. It certifies the BLS and Corona keys via ML-DSA signatures stored at validator registration (not per-block). If BLS is broken, the ML-DSA certificates prove which validator owned which BLS key.

Conclusion

Triple-proof quantum finality provides defense-in-depth against classical, quantum, and future adversaries. By separating speed (BLS), identity (ML-DSA), and privacy (Corona) into independent proof paths unified by Groth16 aggregation, we achieve post-quantum consensus at 248 bytes per epoch — comparable to classical-only systems — while supporting 1ms block times on commodity hardware.

The mechanism is deployed on the Lux Network with 100+ validators and multiple application chains. Measured QuasarCert verification is ∼2–5 ms CPU, dominated by Groth16's 3 pairings on BLS12-381 [7].

References

[1] NIST, "FIPS 204: Module-Lattice-Based Digital Signature Standard (ML-DSA)," 2024.

[2] NIST, "FIPS 203: Module-Lattice-Based Key-Encapsulation Mechanism Standard (ML-KEM)," 2024.

[3] NIST, "FIPS 205: Stateless Hash-Based Digital Signature Standard (SLH-DSA)," 2024.

[4] Lux Industries, Inc., "Corona: Lattice-Based Threshold Ring Signatures for Anonymous Consensus," 2025.

[5] Ben-Sasson et al., "Scalable, transparent, and post-quantum secure computational integrity," IACR ePrint 2018/046.

[6] Boneh, Lynn, Shacham, "Short Signatures from the Weil Pairing," Journal of Cryptology, 2004.

[7] Lux Industries, "Quasar Consensus — Measured Benchmarks," 2026. Apple M1 Max, darwin/arm64, Go 1.26.1. https://github.com/luxfi/consensus/blob/main/BENCHMARKS.md.