Lux MPTC Experimental Evaluation

Zach Kelling

Lux Industries

2026-03-03

Abstract

We present Lux, a hybrid post-quantum consensus architecture for DAG-native chains. Lux separates fast operational finality from durable post-quantum finality. Validators emit Photons over a Nebula DAG substrate; classical BLS aggregates form Beams for low-latency confirmation, while ML-DSA attestations and Pulsar lattice-threshold Pulses provide post-quantum finality over Nebula roots. Prism binds all certificate lanes to the same transcript, and Quasar reaches Horizon finality when the bound lanes verify.

This document. The experimental evaluation summary for the NIST MPTC submission. Detailed numbers, methodology, hardware, and reproduction commands live in the companion benchmark report [2]; this document summarizes the headline results and points at the full report.

1 Headline numbers

All numbers on Apple M1 Max single core, $n = 21$, freeze tag `v0.1.0-rc1-pq-consensus-freeze`. Full reproduction in [2].

1.1 Sign / verify per scheme

Scheme	Sign	Verify
BLS12-381 (single)	350 µs	820 µs
BLS aggregate (100 signers)	—	875 µs
ML-DSA-65	495 µs	250 µs (cached: 3 µs)
Ed25519	—	205 µs (cached: 1.1 µs)
secp256k1	—	658 ns
Pulsar Sign1 (per party, $K = 21$)	185 ms	—
Pulsar Sign2 (per party, $K = 21$)	22 ms	—
Pulsar Verify	—	9 ms
Lens Ed25519 (per party)	110 µs sign	320 µs verify

1.2 Reshare benchmarks (LSS)

LSS-Pulsar resharing $(t, n) \to (t, n)$ at sizes $\{3, 7, 21, 64, 128\}$, local-loop simulation:

n	Pulsar reshare (ms)	Lens reshare (ms)
3	360	0.7
7	570	1.8
21	960	5.2
64	2440	16.5
128	4730	33.0

Lens is $\sim$ 100–200$\times$ faster per-party. Lens is classical (curve discrete log); Pulsar is post-quantum (Module-LWE). The trade-off is the production reason both kernels exist.

1.3 Activation cert latency

Pulsar activation cert at $K = 21$:

Phase	Time
Sign1 + Round1Pre (local-loop)	480 ms
Sign2 + Combine (local-loop)	280 ms
Verify	9 ms
Total local-loop	$\sim$ 760 **ms**
Total cross-WAN p99	$\sim$ 880 **ms**

1.4 Warp 2.0 envelope verify

Destination-side verify latency, $n = 21$:

Path	Time
Beam-only	$\sim$ 900 µs
Beam + Pulse	$\sim$ 10 ms
Full Horizon (Beam + ML-DSA + Pulse)	$\sim$ 15 ms
Beam + Groth16-wrapped ML-DSA + Pulse	$\sim$ 12 ms

1.5 Prism composition verify

Consensus-side full Horizon verify on a Lux mainnet bundle:

Step	Time
Transcript binding check + τ compute	17 µs
Beam aggregate verify ($n = 21$)	875 µs
ML-DSA cert set verify (21, parallel)	5.3 ms
Pulse verify	9 ms
Validator-set + Nebula linkage check	38 µs
Total Prism verify	$\sim$ 15 **ms**

1.6 Cross-WAN $K = 21$ Pulse latency

Phase	Median (ms)	p99 (ms)
Sign1 broadcast (1 WAN crossing)	240	360
Sign1 verify + Round1Pre (local)	220	280
Sign2 broadcast (1 WAN crossing)	240	360
Sign2 verify + Combine (local)	250	320
Total $K = 21$ Pulse	$\sim$ 950	$\sim$ 1320

Lux mainnet bundle interval is $\sim$ 3 s. Pulse latency at p99 is $\sim$ 1.3 s — comfortably under bundle interval. Chain reaches Horizon-final on every bundle in steady state.

1.7 Pulse latency projections at $K = 64$, $K = 128$

K	Median (s)	p99 (s)
21	0.95	1.3
64	1.7	2.3
128	3.0	3.9

At $K \geq 100$, p99 approaches the bundle interval; grouped Pulsar (smaller per-group k, multiple groups in parallel) is the right answer. Open systems-research: closed-form trade-off between G, k, f, and per-group latency under WAN churn.

1.8 GPU acceleration headroom

Projected Pulsar Round 1 GPU speedup over CPU:

K	CPU (ms)	GPU projected (ms)	Speedup
3	95	110	0.9×
21	185	75	2.5×
64	380	95	4.0×
128	740	130	5.7×

Projection from FHE-class NTT throughput on Apple M1 Max ([1]: 2.1 M poly muls/s). GPU production not enabled on the freeze tag; CPU path is the production default.

2 Methodology

Hardware. Apple M1 Max (10 cores, performance + efficiency), Apple M2 Ultra (24 cores), AMD EPYC 7763 (64 cores). Runs on the M1 performance core; throttling monitored via `powermetrics` and runs that triggered throttling are discarded.

Software. Go 1.26.1; `CGO_ENABLED=0` for cross-compile, native otherwise. Build / test gate: `GOWORK=off go test -count=1 -short -timeout 300s ./...` from each canonical repo.

Statistics. Median of 10 runs at `benchtime=1s`. Where p99 is reported, 1000-run distribution. Cross-WAN measurements use the production Lux mainnet 5-region deployment (US-East, US-West, EU-West, AP-South, AP-East); WAN RTT distribution is sampled via the operator monitoring stack.

3 Reproducibility

Every table above is reproducible from the freeze tag using the listed commands in [2]. KAT regen (`pulsar/scripts/regen-kats.sh`) produces byte-equal output across runs. Fuzz harnesses return zero panics in 60 s.

4 Comparison to claims in older Lux drafts

Older Quasar drafts cited a "357 µs epoch finality" figure that does not match any measured operation in the current code. The honest measured numbers are the ones in this report. Closest real candidates to 357 µs: BLS single keygen (350 µs), ML-DSA-65 sign (495 µs); neither corresponds to "epoch finality."

The full Quasar Horizon verify is $\sim$ 15 ms CPU; the cross-WAN $K = 21$ Pulse latency is $\sim$ 1–1.3 s. These are the numbers to cite.

5 Pointers

For the full benchmark report with reproduction commands and per-platform numbers, see [2]. For the underlying threshold-cryptography analyses, see the LP-073 paper [3], the LSS paper [6], the Quasar Horizon paper [4], and the Warp 2.0 paper [5].

References

[1] Zach Kelling. FHE benchmarking, cost models, and operational SLOs. Lux Industries, Inc.; `papers/lux-fhe-benchmarks/`, 2025.

[2] Zach Kelling. Lux post-quantum consensus benchmarks. Lux Industries, Inc.; `papers/lux-pq-consensus-benchmarks/`, 2026.

[3] Zach Kelling. Pulsar: Dynamic lattice threshold signatures for permissionless validator sets. Lux Industries, Inc.; `papers/lp-073-pulsar/`, 2026.

[4] Zach Kelling. Quasar horizon: Hybrid post-quantum finality over dag-native nebula roots. Lux Industries, Inc.; `papers/lux-quasar-horizon/`, 2026.

[5] Zach Kelling. Warp 2.0: Pulse-backed cross-chain source finality. Lux Industries, Inc.; `papers/lux-warp-v2/`, 2026.

[6] Vishnu J. Seesahai and Zach Kelling. LSS: A universal lifecycle framework for threshold cryptographic protocols. Lux Industries, Inc., 2026.