P3Q Corona Kind: Module-LWE Threshold Verify on the Unified Precompile Slot

Zach Kelling

Lux Industries

2026-06-03

Abstract

LP-218 introduces P3Q at EVM slot `0x012205` as the PQ-verifier family for rollup-batch authentication, specifying the unified kind-byte dispatch ABI and the first verifier kind $0x01$ (Pulsar / FIPS-204 ML-DSA-65 threshold). This paper specifies *kind* $0x02$ *Corona*: a Module-LWE threshold verifier built on the two-round Corona construction (eprint 2024/1113). The shared P3Q framework — the unified slot, the kind-byte dispatch ABI, and the `P3QVerifier` registration interface — is specified in LP-219; this paper gives Corona's wire layout and its gas calibration (5 000 000 gas). Corona reduces to Module-LWE / ring-structured lattice hardness, independent of Pulsar's Module-LWE (LP-223) and Magnetar's hash-collision (LP-222) assumptions, and is the Module-LWE parameter-diversity leg of the LP-217 PQ-strict posture. We also place Corona in the DEX rollup settlement pipeline: per-market fill batches roll up to the Z-Chain and are verified on a two-rung ladder — a kind $0x01$ ML-DSA-65 attestation rung (measured ∼4714 verifies/s) and a `starkfri` STARK/FRI validity rung (LP-221, measured ∼986/s on a toy AIR) — with Corona as the optional Module-LWE second leg for value-bearing markets and bundled $K \times V \times$ markets settlement. GPU dispatch is projected, not measured; the on-chain path reserves Corona's dispatch entry and returns `abiFalse` via `ErrKindNotImplemented` until its threshold library lands.

Contents

1 Introduction

P3Q (LP-219) is the unified-slot post-quantum verifier framework at EVM precompile slot `0x012205`: one slot, one ABI, and a leading `kind` byte that dispatches into a per-kind threshold-signature verifier core. This paper specifies *kind* $0x02$ *Corona*, the Module-LWE member of the P3Q family.

For the slot allocation, the kind-byte dispatch ABI, the `P3QVerifier` Go interface that registers each kind, and the cross-family composition into LP-217 cert profile modes, see LP-219. The sibling kinds are specified in LP-223 (Pulsar / rank-k Module-LWE) and LP-222 (Magnetar / hash-based FIPS-205 SLH-DSA).

Corona reduces to the hardness of Module-LWE / ring-structured lattices — built on the two-round Corona threshold construction (eprint 2024/1113) — which is independent of the Module-LWE and hash-collision assumptions its sibling kinds rest on. It is the Module-LWE parameter-diversity leg of the LP-217 PQ-strict posture and is calibrated at 5 000 000 gas, the cost cliff that buys cross-family lattice diversity. Performance numbers are tagged **measured** or **projected**; the on-chain path reserves Corona's dispatch entry and returns `abiFalse` via `ErrKindNotImplemented` until its threshold library lands, and GPU dispatch is projected, not measured.

2 Corona Kind (kind = $0x02$)

2.1 Slot and ABI

`0x012205` with leading `kind=0x02`. Same physical precompile address as Pulsar (kind $0x01$) and Magnetar (kind $0x03$); dispatch resolves to the Corona `P3QVerifier` implementation.

```
// EVM precompile at slot 0x012205, kind = 0x02
bool ok = P3Q.verify(
    0x02,                       // Corona kind
    coronaSig,                  // serialized Corona threshold sig
    messageHash,                // H(prev_root || new_root || batch_hash)
    groupPubKey                 // Corona group public key
);
```

2.2 Underlying scheme

Corona [2] is the Lux deployment of a two-round lattice threshold signature scheme building on the Ringtail line [1]. Hardness reduces to Module-LWE over the ring $R_q = \mathbb{Z}_q[X]/(X^{256}+1)$ at the Ringtail parameter set (module matrix $A \in R_q^{8\times 7}$, 48-bit prime q), via the standard two-round distributed signing protocol.

Corona's signature is *not* byte-equal to any NIST standard. Both Corona and Pulsar (FIPS-204 ML-DSA-65) are Module-LWE schemes; they differ in *parameter regime and codebase*, not in hardness family. Corona uses the Ringtail-derived parameter set (∼33 KB signatures, NIST Cat-1) with Module-SIS polynomial commitments; Pulsar uses the ML-DSA-65 tuning (∼3.3 KB, Cat-3, byte-equal to FIPS-204). Pairing them gives parameter- and implementation-level diversity — a break specific to one parameter set or codebase need not hit the other — but *not* family-disjoint hardness: a structural break of Module-LWE would threaten both. Cross-family independence on the signature lane comes only from the hash-based Magnetar kind (kind $0x03$, SLH-DSA).

2.3 Wire format

The mode byte is reserved at `0x01`; future Corona parameter set additions (a higher-security Module-LWE variant) get new mode bytes without changing the wire-format skeleton.

Offset	Size	Field
0	1	`kind = 0x02`
1	1	`mode = 0x01` (reserved Corona parameter set)
2	4	`sigLen` (BE32)
6	`sigLen`	`coronaSig` (~33 052 B)
6+*sigLen*	4	`pkLen` (BE32)
10+*sigLen*	`pkLen`	`groupPubKey`
10+*sigLen*+*pkLen*	32	`messageHash`

Table 1: Corona-kind wire format. Sig size is the dominant cost ($\sim$ 33 KB); the rest mirrors the Pulsar layout to keep parsing uniform across kinds.

2.4 Verifier core

The Go reference implementation is planned at `lux/standard/precompiles/p3q_family/corona.go` and will wrap `luxfi/corona/sign.Verify`. The core is bespoke – not FIPS-204 byte-equal – because Corona is a separate Module-LWE construction (Ringtail-derived), not a NIST standard.

Status as of 2026-06-04: the verifier core is not yet wired into the P3Q dispatch table. The reference `p3q.Run` routes `kind=0x02` to `abiFalse` via the `ErrKindNotImplemented` path:

```
case KindCorona, KindMagnetar:
    // Reserved kinds -- verifier not yet wired. Return abiFalse
    // ...
    return abiFalse(), remaining, nil
```

Solidity callers MAY emit `P3Q.verify(0x02, ...)` from network genesis; the call charges the per-kind gas (see below) and returns `false` until the verifier lands. This is the LP-218 "no revert on cryptographic failure" contract: `false` on the return word is the universal signal for "not verified for any reason," and reserved-but-not-wired joins "malformed input" and "FIPS reject" under that umbrella.

2.5 Gas cost

5 000 000 gas – $\sim$100$\times$ the Pulsar kind's 50 000. Returned by the Corona `P3QVerifier.GasCost()` entry. Why the disparity?

1. **CPU verify cost is heavier.** The Corona verifier requires NTT inverse over the Module-LWE ring + polynomial multiplication mod q over the larger Ringtail parameter set (~33 KB sig vs Pulsar's 3.3 KB).
2. **Calibration uses the CPU verify cost as the baseline.** LP-218's calibration discipline assumes a CPU-class verifier – validators are not required to have Blackwell GPUs. Under that assumption, Corona's effective per-validator verify cost is the parity audit's measured $\sim$1.6 ms CPU, which is approximately 141$\times$ the Pulsar CPU verify time (~80 µs). The gas charge rounds to 100$\times$ Pulsar's for clean accounting: 5 M gas.
3. **Block-budget headroom.** At 12 M block gas and 5 M per Corona verify, the limit is $\sim$2 Corona verifies per block – this is the bottleneck for very high- density PQ-strict rollup throughput. The cost cliff is intentional; the LP-217 mode ladder uses it to price cross-family insurance correctly.

Operators with all-Blackwell validator sets may petition for a **Blackwell-discounted Corona gas charge** in a future amendment; LP-220 does not codify it. The conservative CPU-baseline gas charge applies until validator-hardware uniformity is established. This is the same calibration discipline LP-218 uses for the Pulsar kind.

Metric	M1 Max	Spark	Source
Per-validator Corona sign at N=64	∼30 ms	∼30 ms	matrix bench (measured) [5]
CPU verify (single sig)	∼1.6 ms	∼1.6 ms	parity audit (measured)
Blackwell sm_120 verify	—	∼15 ms	projected (LP-203 kernel) [3]
Blackwell sm_120 batched (N=64)	—	∼50 ms aggregate	projected

Table 2: Corona kind performance. Sign and CPU-verify times are measured per the parity audit; Blackwell verify timings remain projections until the GPU dispatch boundary closes (see LP-223).

2.6 Performance

2.6.1 Scaling exponent

The parity audit measured Corona-leg signing cost growing super-linearly in N. The measured exponent across $N \in \{16, 32, 64\}$ is approximately $T(N) \approx c \cdot N^{1.5–1.7}$, depending on the host platform. This is not an $O(N)$ scheme and should not be described as such in downstream documents; the Module-LWE distributed signing protocol's full-rank check and Gaussian-hash PRF carry costs that grow faster than linearly in the participant count. The measured exponent is the operative number for sequencer capacity planning, not the naive-amortization figure.

2.7 Use case

A rollup that wants **intra-lattice parameter and implementation diversity** configures its sequencer to sign each batch with both a Pulsar threshold sig AND a Corona threshold sig — two Module-LWE schemes at distinct parameter sets and codebases. The rollup contract on Z-Chain calls `P3Q.verify(0x01, ...)` and `P3Q.verify(0x02, ...)` at the same precompile address `0x012205` in sequence inside its `validateBatch` function; both must return `true` for the batch to be accepted.

A break specific to Pulsar's parameter set or codebase need not compromise Corona's, so the rollup retains a valid PQ leg under an implementation- or parameter-localized failure. This is *not* family-disjoint hardness: a structural break of Module-LWE itself would threaten both legs. For cross-family insurance the rollup must add the hash-based Magnetar kind (`P3Q.verify(0x03, ...)`). The cost is ∼5 050 000 gas per batch (Pulsar 50 000 + Corona 5 000 000). Acceptable for value-bearing rollups (custody, treasury, regulated securities); prohibitive for high-volume rollups (gaming, social).

This is the LP-217 `PQ-strict` posture made operationally callable.

3 Application: DEX Rollup Settlement to Z-Chain

The DEX matching path (Lux Lightspeed DEX [6]) fills orders at hundreds of millions per second per GPU on the D-Chain, far above any single chain's finality rate. To anchor that throughput trustlessly, per-market fill batches roll up to the Z-Chain and are verified there through the P3Q family at slot `0x012205`. This section places Corona (kind $0x02$) in that settlement pipeline; the measured verify rates belong to the sibling kinds, and Corona is the optional Module-LWE parameter-diversity leg.

3.1 Two verification rungs

DEX rollup settlement uses a two-rung ladder, both callable at the same P3Q slot:

1. **Attestation rung (venue-trust).** The sequencer signs each per-market batch with a P3Q kind $0x01$ (Pulsar / FIPS-204 ML-DSA-65) threshold signature. The Z-Chain contract

calls `P3Q.verify(0x01, ...)` to confirm the batch was produced by the venue's threshold quorum. This rung trusts the venue's signer set but is cheap: the ML-DSA-65 verify path is measured at ∼4714 verifies per second.

2. **Validity rung (trustless).** For trust-minimized settlement the sequencer additionally produces a STARK/FRI validity proof of correct matching, verified by the `starkfri` verifier (LP-221 [4]) — a cSHAKE256 / Goldilocks PQ Plonky3-fork verifier. This rung proves the fills are correct without trusting the signer set. The current verifier is measured at ∼986 proofs per second against a **toy AIR**; this figure is the operative number and is explicitly not a production-circuit measurement.

The attestation rung is fast and venue-trusting; the validity rung is slower and trustless. A venue chooses the rung (or runs both) per its trust requirements.

3.2 Where Corona fits

Corona (kind $0x02$) is the *Module-LWE cross-family insurance leg* for the attestation rung, not a separate settlement mechanism. A value-bearing DEX rollup that wants independence from the Module-LWE assumption underlying ML-DSA-65 signs each batch with *both* a Pulsar (kind $0x01$) and a Corona (kind $0x02$) threshold signature and requires both `P3Q.verify` calls to return `true`, exactly as in Section 2's use case. A Module-LWE break that compromises the Pulsar attestation does not compromise the Corona attestation; the rollup's settlement remains cryptographically anchored.

The cost discipline of Section 2 applies unchanged: Corona's 5 000 000 gas makes the second leg appropriate for value-bearing markets (custody, treasury, regulated-securities venues) and prohibitive for high-volume markets, which settle on the kind $0x01$ attestation rung alone (optionally backed by the starkfri validity rung). As noted throughout this paper, Corona's GPU dispatch is projected and its verifier core is not yet wired; the DEX settlement path that needs a second lattice family today uses the measured kind $0x01$ rung, with Corona reserved at its dispatch entry until its threshold library lands.

3.3 Bundled settlement: $K \times V \times$ markets

Per-fill, per-venue verification does not scale: at the attestation-rung's ∼4714 verifies per second, a chain that verified one signature per fill would cap settlement far below the matcher's fill rate. The settlement model therefore bundles. A single Z-Chain settlement object aggregates K signer attestations across V venues across many markets into one verification event:

- K **signers.** The threshold quorum per venue contributes K partial signatures aggregated into one threshold signature that verifies in a single `P3Q.verify` call.
- V **venues.** Independent venues' batches are aggregated into one settlement object rather than verified one venue at a time.
- **Many markets.** Each venue's per-market arenas (Lux Lightspeed DEX) contribute their batch roots to the bundle, so a single verification covers many books.

Because one verification amortizes over $K \times V \times$ markets fills, the effective settled-fill rate at fleet scale is **projected** to exceed 10^9 fills per second of finality — this is a fleet-aggregation projection, not a single-verifier measurement. The per-verifier measured rates (∼4714/s attestation, ∼986/s validity on a toy AIR) are the operative numbers; the $>10^9$ figure is what those rates compose to once the bundle factor $K \times V \times$ markets is applied across a multi-node Z-Chain settlement fleet. No single P3Q verifier reaches that rate.

References

[1] Cecilia Boschini, Darya Kaviani, Russell W. F. Lai, Giulio Malavolta, Akira Takahashi, and Mehdi Tibouchi. Ringtail: Practical two-round threshold signatures from LWE. Cryptology

ePrint Archive, Paper 2024/1113; IEEE Symposium on Security and Privacy (S&P) 2025, 2024. Academic two-round lattice threshold (IACR ePrint 2024/1113, IEEE S&P 2025). Not a Lux artifact; do not relabel as "Corona" or "Pulsar".

[2] Lux Industries. Corona: Two-round Ring-LWE threshold signatures. https://github.com/luxfi/corona, 2026. Lux's own threshold scheme: Ring-LWE (= Module-LWE at rank 1), two-round. Builds on the Ringtail line ([1]); cited via this Lux artifact, never via the academic paper. Q-Chain Feldman/Pedersen DKG.

[3] Lux Industries. Lp-203: Gpu-native verify, 2026.

[4] Lux Industries. Lp-221: starkfri — cshake256 / goldilocks pq plonky3-fork stark/fri verifier, 2026.

[5] Lux Industries. Quasar 4-Scheme Matrix Bench – m1 max + dgx spark gb10, 2026. `lux/threshold/protocols/parity/parity_test.go`, run 2026-06-04. Mean of N=$\{16, 32, 64\}$ across 50 rounds.

[6] Lux Industries, Inc. Lux lightspeed dex: On-chain order book with sub-second finality. Lux Technical Report.