Corona: Two-Round Module-LWE Threshold Signatures for Public Permissionless Chains

Zach Kelling

Lux Industries

2024

Abstract

We present Corona, a two-round Module-LWE threshold signature scheme designed for public, permissionless blockchain consensus. Corona enables any t-of-n subset of a validator set to collaboratively sign blocks without any single party ever holding the full secret key, while providing security against quantum adversaries. The scheme operates in two rounds: a message-independent precomputation and commitment round, followed by a signing round, with a non-interactive combine producing the final aggregate signature. Corona is Lux's first-party fork of Ringtail (Boschini et al., IACR ePrint 2024/1113). It builds on the Module-LWE assumption over the cyclotomic ring $R_q = \mathbb{Z}_q[X]/(X^N + 1)$ with $N = 256$ and a 48-bit prime modulus $q = \texttt{0x1000000004A01}$, instantiated as a module of dimension $M = 8 \times N' = 7$: the public key is a matrix relation $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e}$ with $\mathbf{A} \in R_q^{M \times N'}$, $\mathbf{s} \in R_q^{N'}$, and $\mathbf{e} \in R_q^M$, with Module-SIS commitments. We prove that the scheme achieves existential unforgeability under chosen-message attacks (EUF-CMA) in the random oracle model, reducing security to the hardness of Module-LWE together with Module-SIS, at NIST security Category 1 ($\sim$128-bit classical, $\sim$130-bit quantum). A Corona threshold signature is approximately 33–35 KB and verifies in approximately 1.6 ms; it is *not* byte-equal to any FIPS standard (that byte-equality is a property of the sibling scheme Pulsar, LP-4450, not of Corona). Corona's distinguishing contribution is making the academic two-round lattice threshold construction work for public, permissionless validator sets that change over time: leaderless distributed key generation with no trusted dealer (Feldman/Pedersen commitments), dynamic resharing across a changing validator set, and on-chain ceremony binding anchored to the Lux Q-Chain. It is the intra-lattice diversity leg of the Aurora certificate profile (Pulsar ‖ Corona) under Quasar, verified by the raw EVM precompile at `0x012206`. Corona is a distinct production artifact derived from Ringtail.

Contents

1 Introduction

Threshold signatures enable a group of n parties to collaboratively sign messages such that any subset of t or more parties can produce a valid signature, but fewer than t parties cannot. In blockchain consensus, threshold signatures aggregate validator votes into compact block certificates, reducing verification from $O(n)$ individual signature checks to a single aggregate verification.

Classical threshold schemes—BLS threshold [2], FROST [3], and CGGMP21 [4]—rely on discrete logarithm or pairing assumptions vulnerable to Shor's algorithm [1]. Post-quantum threshold signatures based on lattice assumptions remain an active area of research. Ringtail is a practical two-round Module-LWE threshold construction; that construction, however, assumes a fixed, known set of parties, which is precisely the assumption a public chain cannot make.

Corona is the first member of the Lux lattice threshold-signature family by development age: a two-round Module-LWE threshold scheme, derived from Ringtail (§7), together with its native verifier. Corona works over the cyclotomic ring $R_q = \mathbb{Z}_q[X]/(X^N + 1)$ with $N = 256$, instantiated as a module of dimension $M = 8 \times N' = 7$: the public key is a matrix relation $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e}$ with $\mathbf{A} \in R_q^{M \times N'}$, $\mathbf{s} \in R_q^{N'}$, and $\mathbf{e} \in R_q^M$. Corona and Pulsar [13] (LP-4450) are *both* Module-LWE threshold schemes; they differ by parameter regime, lifecycle, and codebase, not by module rank. The design prioritizes:

- **Low round complexity**: 2 rounds (1 message-independent precomputation and commitment + 1 signing), then a non-interactive combine. Two rounds is optimal for any dealer-free threshold scheme.
- **Compact signature**: ∼33–35 KB, acceptable for block certificates.
- **Fast verification**: ∼1.6 ms (one matrix relation check and one hash evaluation).
- **Standard assumptions**: Module-LWE with Module-SIS commitments, Category 1 parameters.
- **Permissionless operation**: leaderless distributed key generation with no trusted dealer, and dynamic resharing across a validator set that changes over time.

Corona's distinguishing contribution is engineering, not a new hardness assumption: it takes the academic two-round lattice threshold structure and makes it run on a *public, permissionless* validator set. Where the academic construction fixes the participant set in advance, Corona supports validators joining and leaving permissionlessly through leaderless distributed key generation (Feldman/Pedersen commitments, no trusted dealer, no party ever holding the full key), dynamic resharing across generations, and on-chain ceremony binding anchored to the Lux Q-Chain.

Contributions.

1. The Corona two-round threshold signature scheme over Module-LWE with Module-SIS commitments (§3).
2. EUF-CMA security proof reducing to Module-LWE and Module-SIS hardness in the random oracle model (§4).
3. Permissionless, leaderless distributed key generation and resharing for validator sets that change over time, anchored to an on-chain ceremony root (§3, §5).
4. Concrete Category 1 parameters with security estimates (§4).
5. Integration as the Aurora-profile intra-lattice diversity leg under Lux Quasar (§5).
6. Benchmarks: ∼33–35 KB signatures, ∼1.6 ms verification (§6).

2 Preliminaries

2.1 The Cyclotomic Ring

Let $R = \mathbb{Z}[X]/(X^N + 1)$ and $R_q = \mathbb{Z}_q[X]/(X^N + 1)$ where N is a power of 2 and q is a prime with $q \equiv 1 \pmod{2N}$ (enabling the number-theoretic transform, NTT). Elements of R_q are polynomials of degree $< N$ with coefficients in $\mathbb{Z}_q$; the ring product is polynomial multiplication reduced modulo $X^N + 1$. For $f \in R_q$ we write $\|f\|_\infty$ for the largest absolute value of its coefficients (centred representatives in $(-q/2, q/2]$), extended entrywise to vectors and matrices over R_q, and D_σ for the discrete Gaussian over R of width σ applied coefficient-wise. Corona operates over a *module* $R_q^{N'}$: the public key, secret, error and signing values are vectors or matrices over R_q, not single ring elements.

Definition 2.1 (Module-LWE)**.** *The (decisional) Module-LWE problem with parameters (N, q, M, N', σ) is: given a uniformly random matrix $\mathbf{A} \xleftarrow{\$} R_q^{M \times N'}$ and $\mathbf{b} = \mathbf{A}\mathbf{s}+\mathbf{e}$, where $\mathbf{s} \xleftarrow{\$} D_\sigma^{N'}$ and $\mathbf{e} \xleftarrow{\$} D_\sigma^{M}$ are short (discrete Gaussian of width σ) and $\mathbf{A}\mathbf{s}$ uses ring multiplication in R_q, distinguish $(\mathbf{A}, \mathbf{b})$ from $(\mathbf{A}, \mathbf{u})$ where $\mathbf{u} \xleftarrow{\$} R_q^M$ [11].*

Remark 2.1 (Module rank)**.** *Module-LWE at module rank one ($N' = 1$, $M = 1$) coincides with Ring-LWE over a single ring element. Corona is* not *instantiated at rank one: it uses a module of dimension $M = 8 \times N' = 7$ (a genuine matrix $\mathbf{A} \in R_q^{8\times 7}$). Pulsar (LP-4450) is likewise a Module-LWE scheme; Corona and Pulsar differ by parameter regime and lifecycle, not by module rank.*

Definition 2.2 (Module-SIS)**.** *The Module-SIS problem with parameters (N, q, M, β) is: given a uniformly random matrix $\mathbf{A} \xleftarrow{\$} R_q^{M \times N'}$, find a nonzero $\mathbf{z} \in R_q^{N'}$ with $\|\mathbf{z}\|_\infty \le \beta$ and $\mathbf{A}\mathbf{z} = \mathbf{0}$ in R_q^M. Corona's commitments are binding under Module-SIS, and a forgery yields a short Module-SIS solution relative to $\mathbf{A}$.*

2.2 Shamir Secret Sharing over R_q

The secret key is a short vector $\mathbf{s} \in R_q^{N'}$. Each component s_j is shared with one degree-$(t-1)$ polynomial over the ring.

Definition 2.3 (Ring-Based Shamir Sharing)**.** *Given a secret vector $\mathbf{s} = (s_1, \ldots, s_{N'}) \in R_q^{N'}$, share each component s_j via $f_j(X') = s_j + \sum_{i=1}^{t-1} a_{j,i}(X')^i \in R_q[X']$, with $a_{j,i} \xleftarrow{\$} R_q$, evaluated at distinct public points $X' = \mathrm{id}_i \in \mathbb{Z}_q$ (the sharing indeterminate X' is distinct from the ring indeterminate X). Party i receives the share $\mathbf{s}_i = (f_1(\mathrm{id}_i), \ldots, f_{N'}(\mathrm{id}_i)) \in R_q^{N'}$.*

Lagrange interpolation over R_q reconstructs $\mathbf{s}$ from any t shares:

$$\mathbf{s} = \sum_{i \in S} \lambda_i \cdot \mathbf{s}_i, \qquad \lambda_i = \prod_{j \in S,\, j \neq i} \frac{\mathrm{id}_j}{\mathrm{id}_j - \mathrm{id}_i} \in \mathbb{Z}_q \tag{1}$$

where the Lagrange coefficients λ_i are scalars in $\mathbb{Z}_q$ (embedded in R_q).

2.3 Corona Parameters

3 The Corona Scheme

Corona is a two-round Module-LWE threshold signature over the module $R_q^{N'}$. The public key is a matrix relation $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e}$ with public matrix $\mathbf{A} \in R_q^{M \times N'}$, secret $\mathbf{s} \in R_q^{N'}$, and error $\mathbf{e} \in R_q^M$. We first describe the leaderless distributed key generation (no trusted dealer), then the two-round signing protocol, then the non-interactive combine and verification. Throughout, $\mathcal{S}$ denotes a signing set with $|\mathcal{S}| \ge t$, and $\lambda_i \in \mathbb{Z}_q$ are the Lagrange coefficients of (1) for the set $\mathcal{S}$.

Parameter	Symbol	Value
Ring degree	N	256 $(\log_2 N = 8)$
Modulus	q	`0x1000000004A01` (48-bit, $q \equiv 1 \bmod 2N$)
Module rows	M	8
Module columns	N'	7
Gaussian width (key gen)	σ_e	6.108
Gaussian width (signing)	σ^*	$2^{37.33}$
Rejection bound	B	$2^{48.6}$
Challenge ternary weight	κ	23
Rounding parameter	ξ	30

Table 1: Corona parameters (NIST Category 1, Module-LWE). The module dimension $M = 8 \times N' = 7$ and the 48-bit prime are the canonical values of the reference implementation [15] (a fork of `luxfi/ringtail`); the detailed sampler and rejection bounds are fixed by that code and its KAT vectors. These parameters are *not* byte-equal to any FIPS standard; FIPS 204 byte-equality is a property of the sibling scheme Pulsar (LP-4450).

3.1 Distributed Key Generation (Leaderless, No Trusted Dealer)

The public matrix $\mathbf{A} \in R_q^{M \times N'}$ is a fixed, publicly derivable parameter (a nothing-up-my-sleeve hash output expanded from the ceremony root), so no party controls it. The secret $\mathbf{s} \in R_q^{N'}$ is generated jointly: each party contributes an additive summand and a verifiable Shamir sharing of that summand, so the group secret is $\mathbf{s} = \sum_i \mathbf{s}^{(i)}$ and no party ever sees $\mathbf{s}$.

Algorithm 1 Corona Distributed Key Generation

Require: Threshold t, parties n, parameters (N, q, M, N', σ_e), ceremony root ρ

$\mathbf{A} \leftarrow \mathrm{Expand}_{R_q^{M \times N'}}(\rho)$ ▷ public matrix, derived (no dealer)
for each party $i \in \{1, \ldots, n\}$ **do** ▷ run in parallel; leaderless
  $\mathbf{s}^{(i)} \xleftarrow{\$} D_{\sigma_e}^{N'}, \quad \mathbf{e}^{(i)} \xleftarrow{\$} D_{\sigma_e}^{M}$ ▷ i's short secret/error summands
  $\mathbf{f}^{(i)}(X') \leftarrow \mathbf{s}^{(i)} + \sum_{\ell=1}^{t-1} \mathbf{a}_\ell^{(i)} (X')^\ell, \quad \mathbf{a}_\ell^{(i)} \xleftarrow{\$} R_q^{N'}$ ▷ Shamir polys over R_q
  Broadcast Feldman/Pedersen commitments to $\mathbf{s}^{(i)}$ and $\{\mathbf{a}_\ell^{(i)}\}$
  Send share $\mathbf{s}_j^{(i)} \leftarrow \mathbf{f}^{(i)}(\mathrm{id}_j)$ privately to each party j
end for
for each party j **do** ▷ verify and aggregate
  abort/identify any party i whose share $\mathbf{s}_j^{(i)}$ contradicts its commitments
  $\mathbf{s}_j \leftarrow \sum_i \mathbf{s}_j^{(i)}$ ▷ j's aggregate share of $\mathbf{s} = \sum_i \mathbf{s}^{(i)}$
end for
$\mathbf{b} \leftarrow \mathbf{A}\mathbf{s} + \mathbf{e}, \quad \mathbf{e} = \sum_i \mathbf{e}^{(i)}$ ▷ public key, recombined from commitments
$\tilde{\mathbf{b}} \leftarrow \mathrm{Round}(\mathbf{b}, \xi)$ ▷ rounded for verification
return $\mathrm{pk} = (\mathbf{A}, \tilde{\mathbf{b}}), \quad$ shares $\{\mathbf{s}_j\}_{j=1}^n$ ▷ group secret $\mathbf{s}$ never reconstructed

The public key $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e}$ is a Module-LWE sample (Definition 2.1): recovering $\mathbf{s}$ from $(\mathbf{A}, \mathbf{b})$ is the Module-LWE search problem. Because $\mathbf{A}$ is derived from the ceremony root and every contribution is committed, the protocol is leaderless and robust against an adversary controlling up to $t-1$ parties; corrupted shares are caught by commitment verification (identifiable abort). Resharing for a changed validator set re-runs Algorithm 1 over the new set, with the group public key $\tilde{\mathbf{b}}$ held fixed across generations (see §5).

3.2 Signing Protocol (2 Rounds)

Round 1 is message-independent and can be precomputed. Each signer commits to a short masking vector $\mathbf{y}_i \in R_q^{N'}$ via $\mathbf{w}_i = \mathbf{A}\mathbf{y}_i$; correlated randomness with pairwise cancellation keeps the aggregate mask short, and per-message MACs authenticate commitments against malicious injection.

Algorithm 2 Corona Signing — Round 1 (Message-Independent)

Require: Party i, signers set $\mathcal{S}$ with $|\mathcal{S}| \geq t$, session id sid

$\mathbf{r}_i \leftarrow \mathrm{PRF}(\mathrm{key}_i, \mathsf{sid} \| i) \bmod D_{\sigma^*}$ ▷ i's short mask $\in R_q^{N'}$
for each party $j \in \mathcal{S} \setminus \{i\}$ **do**
  $\mathbf{r}_{i,j} \leftarrow \mathrm{PRF}(\mathrm{seed}_{i,j}, \mathsf{sid})$ ▷ pairwise correlated randomness $\in R_q^{N'}$
end for
$\mathbf{y}_i \leftarrow \mathbf{r}_i + \sum_{j:j>i} \mathbf{r}_{i,j} - \sum_{j:j<i} \mathbf{r}_{j,i}$ ▷ masked randomness, pairwise cancellation
$\mathbf{w}_i \leftarrow \mathbf{A}\mathbf{y}_i$ ▷ commitment (matrix-vector product)
for each $j \in \mathcal{S} \setminus \{i\}$ **do**
  $\mathrm{mac}_{i \to j} \leftarrow \mathrm{HMAC}(\mathrm{MACkey}_{i \to j}, \mathsf{sid} \| \mathbf{w}_i)$
end for
broadcast $(\mathbf{w}_i, \{\mathrm{mac}_{i \to j}\}_{j \in \mathcal{S}})$

Algorithm 3 Corona Signing — Round 2 (Message-Dependent)

Require: Party i, message μ, Round 1 data $\{(\mathbf{w}_j, \mathrm{mac}_j)\}_{j \in \mathcal{S}}$

for each $j \in \mathcal{S} \setminus \{i\}$ **do**
  Verify $\mathrm{mac}_{j \to i}$ using $\mathrm{MACkey}_{j \to i}$; **abort** (identify j) if it fails
end for
$\mathbf{w} \leftarrow \sum_{j \in \mathcal{S}} \lambda_j \cdot \mathbf{w}_j$ ▷ combined commitment $\in R_q^M$
$c \leftarrow H(\mathbf{A}, \tilde{\mathbf{b}}, \mathbf{w}, \mu)$ ▷ challenge: ternary polynomial in R_q, weight κ
$\mathbf{z}_i \leftarrow \mathbf{y}_i + c \cdot \lambda_i \cdot \mathbf{s}_i$ ▷ partial response (ring multiplication)
if $\|\mathbf{z}_i\|_\infty > B$ **then** ▷ Fiat–Shamir-with-aborts rejection sampling
  abort and restart with a fresh session id
end if
broadcast $\mathbf{z}_i$

3.3 Combine and Verification

The combine is non-interactive: any party or external aggregator sums the partial responses into a single response vector

$$\mathbf{z} \;=\; \sum_{i \in \mathcal{S}} \lambda_i \cdot \mathbf{z}_i \;\in\; R_q^{N'},$$

and outputs the signature $(c, \mathbf{z})$. By linearity and the Lagrange reconstruction of (1), $\mathbf{z} = \mathbf{y} + c \cdot \mathbf{s}$ where $\mathbf{y} = \sum_i \lambda_i \mathbf{y}_i$ masks the short secret $\mathbf{s}$, so $\mathbf{A}\mathbf{z} - c \cdot \mathbf{b} = \mathbf{A}\mathbf{y} - c \cdot \mathbf{e} \approx \mathbf{w}$.

Verification of $(c, \mathbf{z})$ against $\mathrm{pk} = (\mathbf{A}, \tilde{\mathbf{b}})$ and message μ checks:

1. $\|\mathbf{z}\|_\infty \leq B'$ (the response is short, for the combined bound B');
2. recompute $\mathbf{w}' \leftarrow \mathbf{A}\mathbf{z} - c \cdot \tilde{\mathbf{b}} \in R_q^M$ and check $c = H(\mathbf{A}, \tilde{\mathbf{b}}, \mathrm{Round}(\mathbf{w}', \xi), \mu)$.

Both checks are one matrix-vector product over R_q plus one hash, giving the $\sim$1.6 ms verification of §6. A successful forgery on a fresh message yields a short $\mathbf{z}^*$ with $\mathbf{A}\mathbf{z}^* \approx \mathbf{w}^* + c^* \cdot \tilde{\mathbf{b}}$, i.e. a short Module-SIS solution relative to $\mathbf{A}$ (Definition 2.2); this is the basis of the reduction in §4.

4 Security Analysis

4.1 Unforgeability

Theorem 4.1 (EUF-CMA Security)**.** *Corona is existentially unforgeable under chosen-message attacks in the random oracle model, assuming the hardness of Module-LWE(N, q, M, N', σ_e) (Definition 2.1) and Module-SIS(N, q, M, β) (Definition 2.2). Concretely, any adversary making Q_H hash queries and Q_S signing queries has advantage:*

$$Adv^{EUF\text{-}CMA} \leq Adv^{MLWE} + Q_H \cdot Adv^{MSIS} + \frac{Q_S}{2^{128}}.$$

Proof. We proceed through a sequence of games.

Game 0: The real EUF-CMA game. The adversary controls $t-1$ parties and interacts with the signing oracle.

Game 1: Replace the public key $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e}$ with a uniformly random $\mathbf{u} \xleftarrow{\$} R_q^M$. The distinguishing advantage between Games 0 and 1 is $\mathrm{Adv}^{\mathrm{MLWE}}$ by Definition 2.1: the pair $(\mathbf{A}, \mathbf{b})$ is exactly a Module-LWE sample and $(\mathbf{A}, \mathbf{u})$ its uniform counterpart.

Game 2: In Game 1 the public key $\mathbf{b} = \mathbf{u}$ is independent of $\mathbf{s}$. We reprogram the random oracle H to extract the forgery. When the adversary outputs a forgery $(c^*, \mathbf{z}^*)$ on a fresh μ^*, validity requires

$$\mathbf{A}\mathbf{z}^* - c^* \cdot \tilde{\mathbf{u}} \;\approx\; \mathbf{w}^* \qquad \text{with } c^* = H(\mathbf{A}, \tilde{\mathbf{u}}, \mathbf{w}^*, \mu^*),$$

i.e. a short $\mathbf{z}^*$ with $\mathbf{A}\mathbf{z}^* \approx \mathbf{w}^* + c^* \cdot \tilde{\mathbf{u}}$. Since $\mathbf{u}$ is uniform and independent of $\mathbf{s}$, producing such a short $\mathbf{z}^*$ is an instance of Module-SIS (Definition 2.2) relative to $\mathbf{A}$: it yields a short nonzero vector annihilated by $\mathbf{A}$. Each of the Q_H oracle queries gives one such extraction opportunity, whence the $Q_H \cdot \mathrm{Adv}^{\mathrm{MSIS}}$ term.

The probability of a challenge collision in H (two queries yielding the same ternary challenge for different messages) is $\leq Q_H^2/|C|$, where the challenge space $|C| = \binom{N}{\kappa}2^\kappa$ is negligible for $\kappa = 23$ and $N = 256$.

Combining the transitions, $\mathrm{Adv}^{\text{Game 0}} \leq \mathrm{Adv}^{\mathrm{MLWE}} + Q_H \cdot \mathrm{Adv}^{\mathrm{MSIS}} + Q_H^2/|C|$. □

4.2 Threshold Security

Lemma 4.2 (Threshold Unforgeability)**.** *An adversary controlling $t-1$ parties and observing Q_S signing sessions cannot forge a signature on a new message. The adversary's view in each signing session is statistically close to a simulation that does not use the honest parties' shares, by the zero-knowledge property of the commitment scheme.*

Proof. In each signing session, the adversary sees the honest parties' commitments $\mathbf{w}_i = \mathbf{A}\mathbf{y}_i$ and responses $\mathbf{z}_i = \mathbf{y}_i + c\lambda_i\mathbf{s}_i$. Given $\mathbf{w}_i$ and $\mathbf{z}_i$, the adversary can compute $c\lambda_i\mathbf{s}_i = \mathbf{z}_i - \mathbf{y}_i$. However, reconstructing $\mathbf{y}_i$ from $\mathbf{w}_i = \mathbf{A}\mathbf{y}_i$ is an instance of Module-SIS (finding a short preimage of $\mathbf{w}_i$ under $\mathbf{A}$, Definition 2.2).

The Fiat–Shamir-with-aborts rejection sampling ensures that $\mathbf{z}_i$ is statistically independent of $\mathbf{s}_i$: the distribution of $\mathbf{z}_i$ conditioned on any $\mathbf{s}_i$ is within statistical distance 2^{-128} of the distribution conditioned on any other $\mathbf{s}_i$, by the properties of discrete Gaussian rejection sampling with width $\sigma^* \gg \|c\lambda_i\mathbf{s}_i\|_\infty$.

Therefore $t-1$ parties' shares reveal no information about the group secret $\mathbf{s}$ (by the Shamir sharing of Definition 2.3), and the signing transcripts reveal no additional information about $\mathbf{s}$ (by the zero-knowledge simulation). The same argument applies to the distributed key generation: an adversary corrupting up to $t-1$ parties learns nothing about $\mathbf{s} = \sum_i \mathbf{s}^{(i)}$ beyond what the public key $\mathbf{b}$ already reveals, since the honest contributions $\mathbf{s}^{(i)}$ remain hidden behind their commitments. □

4.3 Concrete Security Estimates

Proposition 4.3 (Security Level)**.** *With the Category 1 parameters $N = 256$, $q \approx 2^{48}$, $M = 8$, $N' = 7$, $\sigma_e = 6.108$ (Table 1), the Module-LWE instance reaches NIST security Category 1: ∼128-bit classical and ∼130-bit quantum security against the best known lattice algorithms (BKZ with sieving, Grover-enhanced sieving).*

Proof. The Module-LWE instance embeds into an LWE lattice of dimension $d = M \cdot N = 8 \times 256 = 2048$. The noise ratio is $\alpha = \sigma_e\sqrt{2\pi}/q \approx 15.3/2^{48} \approx 2^{-44.1}$. The primal lattice attack via BKZ with sieving has classical cost

$$T_{\text{classical}} \approx 2^{0.292\,\beta + o(\beta)},$$

where the blocksize β is the smallest value at which BKZ recovers the secret for the (d, q, σ_e) instance; the geometric-series-assumption estimate $\delta^{2\beta-d} \cdot q^{d/\beta} = \sigma_e$ gives $\beta \approx 430$ and $T_{\text{classical}} \approx 2^{142}$. For quantum attacks (Grover-accelerated sieving) the exponent constant drops to 0.265, giving $T_{\text{quantum}} \approx 2^{130}$ at the same blocksize. The Module-SIS commitments are parameterised over the same ring and modulus and are no easier than the Module-LWE instance. Corona is therefore a Category 1 (Module-LWE) scheme. It is *not* byte-equal to any FIPS standard—byte-equality with FIPS 204 ML-DSA is a property of the sibling scheme Pulsar (LP-4450), not of Corona. □

5 Integration with Lux Quasar

Corona is composed under the Lux Quasar consensus engine as the intra-lattice diversity leg of multi-leg finality certificates. A Quasar certificate profile bundles independently verifiable signature legs, each of which must validate for finality:

1. **BLS leg**: classical BLS12-381 threshold signatures provide the fast (1-round) aggregation lane.
2. **Pulsar leg**: threshold ML-DSA over Module-LWE (FIPS 204 byte-equal, LP-4450) is the post-quantum floor.
3. **Corona leg**: this scheme—two-round Module-LWE, Ringtail-derived—is the intra-lattice diversity leg, profile-selectable alongside Pulsar.

The **Aurora** certificate profile (LP-4900) is Pulsar ‖ Corona; the **Polaris** profile (LP-4910) is Pulsar ‖ Corona ‖ Magnetar, adding the hash-based Magnetar leg (SLH-DSA, LP-4540).

Diversity caveat. Corona and Pulsar are *both* Module-LWE constructions; they differ by parameter regime, lifecycle, and codebase, not by module rank. The Pulsar ‖ Corona composition of Aurora is therefore *intra-lattice* defence-in-depth—different code paths, different parameter sets, different rejection samplers—and is honestly *not* family-disjoint hardness: a structural break of the cyclotomic-ring module lattice problem could in principle touch both legs. True cross-family diversity is provided by the hash-based Magnetar leg in the Polaris profile, whose security rests on hash (SLH-DSA, FIPS 205) rather than lattices. Aurora buys implementation and parameter diversity; Polaris buys cryptographic-family diversity.

5.1 On-Chain Verification

A Corona threshold signature is verified by the native EVM precompile at the raw slot `0x012206` (LP-0120): the precompile checks the aggregated two-round signature against the group public key in ∼1.6 ms. Under the P3Q unified rollup-batch verifier (LP-218 / LP-220) Corona is **kind byte** `0x02` at slot `0x012205`; the raw `0x012206` path remains for non-rollup-batch use.

5.2 Permissionless Key Lifecycle

Corona's distributed key generation (Algorithm 1) runs leaderless over the *current* validator set, with no trusted dealer, anchored to the Q-Chain ceremony root ρ. As validators join and leave permissionlessly, the key material is reshared across generations: each generation re-runs the DKG over the new set while holding the group public key fixed for cross-generation verification, and snapshots are committed on-chain. Rotation is rate-limited (a minimum dwell between rotations) and force-rotated on an upper bound, so a churning validator set cannot exhaust the rotation machinery. This permissionless, on-chain-anchored lifecycle—rather than any change to the underlying two-round lattice algebra—is what adapts the academic threshold construction to a public chain.

6 Empirical Evaluation

6.1 Benchmarks

Operation	**Time**	**Size**
Distributed key generation (3 parties)	1.2s	— (no dealer; shares local)
Round 1 (per party, precomputable)	85ms	commitment $\mathbf{w}_i \in R_q^M$
Round 2 (per party)	120ms	response $\mathbf{z}_i \in R_q^{N'}$
Total signing (2-of-3)	0.6s	—
Verification	1.6ms	—
Signature size	—	∼33–35 KB

Table 2: Corona performance benchmarks (Module-LWE, Category 1; signature size and verification time per LP-0120).

6.2 Geographic Distribution

With 5 parties across US-East, US-West, EU-West, Asia-Pacific, and South America (100–250 ms inter-region latency), end-to-end signing for a 3-of-5 threshold was dominated by network round-trips rather than computation, comfortably within the finality-certificate cadence. Because Round 1 is message-independent, it can be precomputed, leaving only the Round 2 round-trip and the non-interactive combine on the critical path.

6.3 Comparison with Classical Threshold Schemes

Scheme	**Sig Size**	**Sign Time**	**Verify**	**PQ Safe**
BLS Threshold	48 B	5ms	2ms	No
FROST	64 B	10ms	1ms	No
CGGMP21	65 B	200ms	<1ms	No
Corona	**∼33–35 KB**	**600ms**	**1.6ms**	**Yes**

Table 3: Threshold signature scheme comparison. Corona is the price of quantum resistance on the same dealer-free, two-round profile as FROST.

Corona's ∼33–35 KB signature is larger than classical schemes by roughly three orders of magnitude, but this is acceptable for block certificates (one signature per block). Corona's signature is an order of magnitude larger than Pulsar's ∼3.3 KB—Pulsar's parameters are tuned

for FIPS 204 byte-equality and a compact encoding, whereas Corona carries the larger Ringtail-derived module and rejection bounds—which is the cost Corona pays for being an independent implementation in the Aurora diversity profile. The signing time is dominated by network latency rather than computation, and the ∼1.6 ms verification is competitive with classical schemes.

7 Related Work

Bendlin et al. [5] introduced lattice-based threshold signatures using NTRU. Boneh et al. [6] proposed threshold signatures from lattice assumptions but with impractical parameters. Cozzo and Smart [7] and Espitau et al. [8] improved practical lattice-based threshold schemes. Corona is Lux's first-party fork of Ringtail [9], a practical *two-round* Module-LWE threshold signature whose construction assumes a fixed, known set of parties. Corona is Lux's own production scheme: it extends Ringtail to public, permissionless validator sets via leaderless distributed key generation, dynamic resharing, and on-chain ceremony binding—engineering that the academic construction does not provide. The module ring R_q itself is the ideal-lattice setting of Lyubashevsky, Peikert and Regev [10], with the module structure of Langlois and Stehlé [11].

Within the Lux lattice threshold family, Corona and Pulsar [13] (LP-4450) are both Module-LWE threshold signatures; they differ by parameter regime, lifecycle, and codebase, not by module rank. Pulsar is the production-tuned variant—threshold ML-DSA-65, byte-equal to FIPS 204, with ∼3.3 KB signatures—while Corona carries the larger Ringtail-derived parameters and the permissionless lifecycle. Magnetar [14] (LP-4540) is the hash-based threshold leg (SLH-DSA, FIPS 205), providing the cross-family diversity that two lattice schemes cannot. Corona, Pulsar and Magnetar are composed by the Aurora and Polaris certificate profiles under Quasar.

8 Conclusion

Corona is the foundational member of the Lux lattice threshold-signature family: a two-round Module-LWE threshold scheme over the module $R_q^{N'}$ (with $\mathbf{A} \in R_q^{8\times 7}$, $R_q = \mathbb{Z}_q[X]/(X^{256}+1)$, $q = \texttt{0x1000000004A01}$), with Module-SIS commitments and Category 1 security (∼128-bit classical, ∼130-bit quantum). It produces ∼33–35 KB signatures that verify in ∼1.6 ms via the native precompile at `0x012206`. Its distinguishing contribution is engineering rather than a new hardness assumption: Corona takes the Ringtail two-round lattice threshold construction and makes it run on a public, permissionless validator set through leaderless distributed key generation with no trusted dealer, dynamic resharing across a changing set, and on-chain ceremony binding anchored to the Lux Q-Chain. Under Quasar, Corona is the intra-lattice diversity leg of the Aurora profile (Pulsar ‖ Corona) and Polaris profile (Pulsar ‖ Corona ‖ Magnetar); the production-tuned, FIPS 204-byte-equal Module-LWE sibling is Pulsar [13] (LP-4450), and the cross-family hash leg is Magnetar [14] (LP-4540). The reference implementation is `github.com/luxfi/corona` [15] (a fork of `luxfi/ringtail`), with machine-checked correctness in `proofs/lean/Crypto/Corona.lean`.

References

[1] P. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM J. Computing*, 1997.

[2] A. Boldyreva, "Threshold signatures, multisignatures and blind signatures based on the gap-Diffie-Hellman-group signature scheme," in *PKC*, 2003.

[3] C. Komlo and I. Goldberg, "FROST: Flexible round-optimized Schnorr threshold signatures," in *SAC*, 2020.

[4] R. Canetti et al., "UC non-interactive, proactive, threshold ECDSA with identifiable aborts," in *CCS*, 2021.

[5] R. Bendlin, S. Krehbiel, C. Peikert, and A. Rosen, "Semi-homomorphic encryption and multiparty computation," in *EUROCRYPT*, 2013.

[6] D. Boneh et al., "Threshold cryptosystems from threshold fully homomorphic encryption," in *CRYPTO*, 2018.

[7] D. Cozzo and N. Smart, "Sharing the LUOV: Threshold post-quantum signatures," in *IMA*, 2023.

[8] T. Espitau, M. Tibouchi, and A. Wallet, "Practical Fiat-Shamir with aborts threshold signatures from lattices," 2024.

[9] G. Boschini, A. Kaviani, R. W. F. Lai, G. Malavolta, A. Takahashi, and M. Tibouchi, "Ringtail: Practical two-round threshold signatures from LWE," IACR ePrint 2024/1113; IEEE S&P 2025.

[10] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," in *EUROCRYPT*, 2010.

[11] A. Langlois and D. Stehlé, "Worst-case to average-case reductions for module lattices," *Designs, Codes and Cryptography*, 2015.

[12] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," *JACM*, 2009.

[13] Lux Core Team, "Pulsar: Module-LWE threshold ML-DSA-65 (FIPS 204 byte-equal)," Lux Proposal LP-4450.

[14] Lux Core Team, "Magnetar: hash-based threshold SLH-DSA (FIPS 205)," Lux Proposal LP-4540.

[15] Lux Core Team, "Corona reference implementation," `github.com/luxfi/corona` (v0.7.6); Lux Proposal LP-4440.