DYTALLIX

Quantum-Secure, Open-Source, Future-Ready

Technical Whitepaper

January 2026

Abstract

Dytallix is a Layer 1 blockchain engineered to withstand the cryptanalytic capabilities of fault-tolerant quantum computers (CRQC). Unlike hybrid transition layers that wrap classical ECC with temporary post-quantum veneers, Dytallix implements a strictly PQC-native stack. The architecture integrates ML-DSA-65 (FIPS 204) for consensus and transaction authorization, ML-KEM-768 (FIPS 203) for transport layer security, and SLH-DSA (FIPS 205) for cold-storage assurance. This document details the specific parameter choices, consensus logic, and virtual machine opcodes that constitute the Dytallix network.

1. The Quantum Imperative: Engineering for the Y2Q Era

The advent of Cryptographically Relevant Quantum Computers (CRQCs) represents the single largest discontinuity in the history of information security. Current digital infrastructure relies almost exclusively on the computational hardness of integer factorization (RSA) and the discrete logarithm problem (ECC/Diffie-Hellman). These mathematical foundations, while robust against classical supercomputers, are fragile against Shor's algorithm running on quantum hardware.

1.1 THE SCOPE OF THE COLLAPSE

The threat is not limited to a specific sector; it is systemic.

- **Cybersecurity:** The Public Key Infrastructure (PKI) that secures the global internet—from TLS handshakes to software signing keys—will be rendered insecure.
- **Software Engineering:** Billions of lines of legacy code assume small key sizes (e.g., 32-byte private keys) and fast verification times. The transition to Post-Quantum Cryptography (PQC) requires re-engineering systems to handle larger keys (kilobytes vs. bytes) and heavier verification logic.
- **System Architecture:** Distributed systems, particularly blockchains, face an existential risk. A quantum adversary could derive private keys from public addresses, forging signatures to drain funds or hijack consensus mechanisms, effectively unwinding the immutable ledger.

1.2 THE "HARVEST NOW, DECRYPT LATER" CRISIS

The threat is not distant; it is retroactive. Adversaries are currently executing "Harvest Now, Decrypt Later" (HNDL) attacks, capturing encrypted traffic today to decrypt it once a CRQC becomes available. This renders any data with a secrecy shelf-life beyond the "Y2Q" horizon (estimated ~2030-2035) effectively compromised *today* if transmitted over classical channels.

1.3 THE DYTALLIX SOLUTION

Dytallix addresses this looming architectural collapse by establishing a strictly PQC-native Layer 1 blockchain. Rather than patching vulnerabilities with hybrid schemes that retain classical weaknesses, Dytallix rebuilds the trust layer from the ground up using NIST-standardized lattice cryptography. By integrating PQC into the consensus, networking, and account models, Dytallix provides a durable infrastructure capable of securing value through and beyond the quantum transition.

2. Cryptographic Primitive Specification

Dytallix establishes a comprehensive cryptographic baseline designed to resist quantum adversaries while maintaining practical performance for decentralized networks. This specification details the selection of standardized, NIST-approved algorithms for digital signatures, key encapsulation, and hashing, ensuring a robust foundation for the entire protocol stack.

2.1 SIGNATURE SCHEME: ML-DSA (MODULE-LATTICE)

Dytallix standardizes ML-DSA-65 (NIST Security Level 3) as the default signature scheme for user accounts and validator voting. This parameter set balances the security equivalent of AES-192 with manageable artifact sizes.

- **Algorithm:** ML-DSA-65 (derived from Dilithium3).
- **Security Assumption:** Hardness of Module-LWE (Learning With Errors) and Module-SIS (Short Integer Solution).
- **Parameter Profile**:
  - Public Key: 1,952 bytes.
  - Private Key: 4,032 bytes.
  - **Signature:** 3,309 bytes.
- P**erformance** (Ref: AVX2 implementation):
  - Signing: ~105 k cycles.
  - Verification: ~97 k cycles.
- **Determinism:** Signatures are generated deterministically (using a pseudorandom function of the message and secret key) to prevent nonce-reuse attacks that could leak private key bits.

2.2 KEY ENCAPSULATION: ML-KEM (MODULE-LATTICE)

Network handshakes and mempool privacy envelopes utilize ML-KEM-768 (NIST Security Level 3).

- **Algorithm:** ML-KEM-768 (derived from Kyber768).
- **Function:** IND-CCA2 secure Key Encapsulation Mechanism.
- **Artifact Sizes**:
  - Public Key: 1,184 bytes.
  - Ciphertext: 1,088 bytes.
  - Shared Secret: 32 bytes (256-bit).

2.3 HASH FUNCTIONS & MERKLEIZATION

- **Primary Hash**: BLAKE3 is used for state trie construction and content-addressing due to its superior throughput on parallel architectures (SIMD).
- **Standard Compliance**: SHAKE256 (FIPS 202) is used specifically within the ML-DSA and ML-KEM internal routines and for generating the final block hash to ensure strict NIST compliance.
- **State Commitment**: The global state is committed via a Sparse Merkle Tree (SMT) optimized for 32-byte keys, mitigating the storage overhead of large PQC public keys by hashing them to standard 256-bit addresses.

2.4 COLD-STORAGE SIGNATURE: SLH-DSA (STATELESS HASH-BASED)

For cold-storage assurance, Dytallix integrates SLH-DSA-SHAKE-192s (NIST Security Level 3) as a complementary signature scheme. This provides diversity from lattice assumptions, relying instead on the quantum-hardness of hash functions (resistant beyond Grover's square-root speedup). Ideal for offline, long-term key storage where signatures are generated infrequently.

- Algorithm: SLH-DSA-SHAKE-192s (derived from SPHINCS+).
- Security Assumption: Pre-image and collision resistance of SHAKE (FIPS 202).
- Parameter Profile:
- Public Key: 48 bytes.
- Private Key: 96 bytes.
- Signature: 16,224 bytes.
- Performance: Signing is compute-intensive (~millions of hash calls), but verification is fast. Deterministic and stateless to avoid nonce issues.
- Use in Dytallix: Optional for air-gapped wallets or archival nodes, ensuring unforgeability even if lattice problems fall to future quantum advances.

3. Mathematical Formalization

Dytallix security relies on the worst-case to average-case reductions of lattice problems over module lattices. The core operations occur within the polynomial ring $R_q = \mathbb{Z}_q[X]/(X^{256}+1)$, where q is a specific prime modulus satisfying $q \equiv 1 \pmod{256}$ to enable efficient Number Theoretic Transforms (NTT).

3.1 HARDNESS ASSUMPTIONS

The protocols rest on two primary computational problems that are believed to be hard even for quantum computers:

- **Module-LWE (Learning With Errors):** Given a matrix $\mathbf{A}$ chosen uniformly from $R_q^{k\times l}$ and a vector $\mathbf{t}$ constructed as:

  $$\mathbf{t} := \mathbf{A}\mathbf{s} + \mathbf{e} \pmod{q}$$

  where the secret vector $\mathbf{s}$ and error vector $\mathbf{e}$are sampled from a small, centered binomial distribution χ, the problem is to distinguish the pair $(\mathbf{A}, \mathbf{t})$ from a pair $(\mathbf{A}, \mathbf{u})$ where $\mathbf{u}$ is uniformly random. This assumption protects the confidentiality of ML-KEM keys.

- **Module-SIS (Short Integer Solution):** Given a matrix $\mathbf{A}$ chosen uniformly from $R_q^{k\times l}$, the problem is to find a non-zero vector $\mathbf{z}$ with a small Euclidean norm such that:

  $$\mathbf{A}\mathbf{z} \equiv \mathbf{0} \pmod{q}$$

  This assumption underpins the unforgeability of ML-DSA signatures.

3.2 SIGNATURE MECHANICS: FIAT-SHAMIR WITH ABORT

The ML-DSA scheme implemented in Dytallix utilizes the "Fiat-Shamir with Abort" paradigm to construct a signature without leaking the secret key $\mathbf{s}$. The process ensures that the distribution of the signature $\mathbf{z}$ is independent of $\mathbf{s}$.

1. **Preparation:** The signer samples a masking vector $\mathbf{y}$ from a distribution with range much larger than the secret key.
2. **Commitment:** Compute $\mathbf{w}_1 = \text{HighBits}(\mathbf{A}\mathbf{y})$ where HighBits drops low-order bits to reduce signature size.
3. **Challenge:** Compute the challenge $c = H(\mu || \mathbf{w}_1)$, where μ is the message digest.
4. **Response:** Compute potential signature $\mathbf{z} = \mathbf{y} + c\,\mathbf{s}$.
5. **Rejection Sampling (The Abort):** The signer checks two conditions:
   - **Norm Check:** If $||\mathbf{z}||_\infty \geq \gamma_1 - \beta$, reject and resample (Ensures $\mathbf{z}$ does not leak $\mathbf{s}$).
   - **Low-Order Check:** The low-order bits of $(\mathbf{A}\mathbf{z} - c\,\mathbf{t})$ must match the bits omitted when computing $\mathbf{w}_1$ in step 2. If either check fails, the signer rejects $\mathbf{y}$ and restarts with a new mask. This non-deterministic loop is made isochronous in the Dytallix implementation to prevent timing side-channels.

4. The Two-Fold Quantum Threat Landscape

The rationale for Dytallix's PQC-native architecture stems from two distinct threat horizons: the immediate risk to confidentiality and the future risk to integrity.

4.1 HARVEST NOW, DECRYPT LATER (HNDL)

This immediate threat targets the confidentiality of data traveling over the wire. Classical Diffie-Hellman (ECDH/RSA) handshakes lack quantum resistance.

- **The Vector**: Adversaries tap fiber lines or compromise intermediate routers to record encrypted traffic (ciphertext).
- **The Failure Mode**: Once a CRQC becomes available (Q-Day), the adversary uses Shor's algorithm to solve the Discrete Logarithm Problem (DLP), recovering the session keys from the recorded handshakes.
- **Dytallix Mitigation**: All P2P traffic and RPC calls use ML-KEM-768 for key agreement. This ensures Post-Quantum Forward Secrecy (PQ-FS). Even if a node's long-term identity key is compromised in the future, past session keys cannot be recovered because they were negotiated using ephemeral lattice-based parameters.

4.2 ACTIVE QUANTUM ATTACKS (POST-CRQC)

Once a CRQC is active, the threat shifts from passive decryption to active forgery.

- **Signature Forgery:** Shor's algorithm reduces the complexity of breaking ECDSA (secp256k1) from sub-exponential time to polynomial time in the bit-length of the modulus. An attacker can derive a private key d from any exposed public key Q.
- **Consensus Hijacking:** In classical PoS chains, validators sign votes with BLS or Ed25519. A quantum attacker could forge votes from high-stake validators, effectively taking control of the chain's finality gadget without owning the stake.
- Dytallix Mitigation:

- Consensus: All validator votes are signed with ML-DSA-65.
- User Funds: Unforgeability reduces to the hardness of Module-SIS, which in turn has worst-case to average-case reductions from lattice problems such as SVP/SIVP. No efficient quantum algorithm is known to solve these problems in polynomial time.

5. Address & Identity Model

To mitigate "Harvest-Now, Decrypt-Later" (HNDL) attacks against long-lived identities, Dytallix enforces a strict Hash-Commit-Reveal pattern for addresses.

5.1 THE DYTALLIX ADDRESS (D-ADDR)

A Dytallix address is a 32-byte BLAKE3 hash of the ML-DSA-65 public key, encoded in Bech32m with the HRP dyt: D-Addr = Bech32m(BLAKE3(ML-DSA-65_PubKey))

- **Pre-Spend Safety:** As long as an address has never spent funds, its public key remains hidden behind a 256-bit hash. Attacks requiring the public key (like Shor's algorithm) are inapplicable during this phase.
- **Post-Spend Security:** Upon the first spending transaction, the public key is revealed to the network. From this point forward, security relies entirely on the strength of the ML-DSA-65 signature scheme.
- **Key Rotation:** Accounts support native key rotation. The state object maps D-Addr -> Current_PubKey_Hash. Users can update the underlying PQC key (e.g., moving from ML-DSA-65 to SLH-DSA) without changing their public address.

6. Consensus Architecture: “Nakamoto-Flow"

Dytallix employs a Nakamoto-style Proof-of-Stake (PoS) mechanism augmented with BFT Checkpoints. This hybrid model provides the liveness of longest-chain rules with the deterministic finality of BFT gadgets.

6.1 THE EPOCH & SLOT STRUCTURE

- Slot Time: 2 seconds.
- **Epoch:** 100 slots (3.33 minutes).
- **Block Proposer:** Selected via a Verifiable Random Function (VRF) hardened against quantum bias. The VRF output determines the "slot leader" based on stake weight.

6.1.1 CRYPTOGRAPHIC SORTITION (FORMAL DEFINITION)

Leader selection is non-interactive and private. A validator v with active stake w_v checks if they are the elected leader for slot t by evaluating a Verifiable Random Function (VRF) against a difficulty threshold τ.

Let σ_v be the validator's secret key and η_t be the epoch entropy. The validator computes the pseudorandom output y:

$$y = \mathrm{VRF}_{sk}(\eta_t \,||\, \mathrm{slot}_t)$$

The validator is the valid proposer if and only if:

$$\frac{H(y)}{2^{256}} < \frac{w_v}{W_{total}}$$

Or equivalently: $H(y) < 2^{256} \cdot \frac{w_v}{W_{total}}$.

This inequality ensures that the probability of election is strictly proportional to stake weight, while the VRF proof π (attached to the block header) allows purely public verification without revealing the long-term secret key.

6.2 BLOCK VALIDATION FLOW

1. **Header Verification:** Nodes check the VRF output and the proposer's ML-DSA-65 signature on the block header.
2. **Batch Verification:** Transactions within the block are verified using AVX2-optimized **parallel batching**. Instead of verifying 5,000 signatures sequentially, the node verifies them in chunks, exploiting the linearity of lattice operations.
3. **Fork Choice Rule:** LMD-GHOST (Latest Message Driven Greediest Heaviest Observed SubTree). Nodes follow the chain with the most accumulated stake-weight in votes.

6.3 FINALITY GADGET (THE CHECKPOINT)

At the end of every epoch, a distinct BFT committee signs a Checkpoint Block**.**

- **Supermajority:** $> \frac{2}{3}$ of total active stake must sign the checkpoint.
- **Signature Aggregation:** Unlike BLS, ML-DSA does *not* support constant-size non-interactive aggregation. Dytallix aggregates signatures into a Bitfield-Compressed Merkle **Proof**. The checkpoint contains a bitfield of signers and a single aggregate root, reducing the on-chain footprint.
- **Finality:** Once a checkpoint is justified and finalized, deep reorgs are impossible.

6.4 FORMAL SAFETY ANALYSIS (BFT QUORUM INTERSECTION)

Dytallix safety protocols rely on the property that two conflicting checkpoints cannot both acquire a supermajority ($> \frac{2}{3}$) of the stake without a significant intersection of malicious validators.

Let S be the total stake of the network. A checkpoint C is considered justified if it gathers a set of votes V_C such that:

$$Weight(V_C) > \frac{2}{3}S$$

Safety Lemma: Assume two conflicting checkpoints C_1 and C_2 at the same epoch height are both justified. Then:

$$Weight(V_{C1}) + Weight(V_{C2}) > \frac{2}{3}S + \frac{2}{3}S = \frac{4}{3}S$$

Since the total stake is S, the intersection of votes must be:

$$Weight(V_{C1} \cap V_{C2}) > \frac{4}{3}S - S = \frac{1}{3}S$$

Thus, at least $\frac{1}{3}$ of the total stake must have equivocated (double-signed). Since the slashing condition (Section 8.1) burns 100% of the stake for double-signing, the cost of finalizing a conflicting chain is precisely $\frac{1}{3}S$, providing a quantifiable crypto-economic security bound.

7. Networking: PQC-Noise Protocol

The P2P layer is a custom implementation of libp2p using a modified Noise Protocol Framework.

7.1 HANDSHAKE PATTERN: NOISE_IK_PQC

Standard Noise protocols (like IK) rely on Diffie-Hellman (DH). Dytallix replaces DH with KEM+Auth.

1. **Initiator (Alice)**:
   - Generates ephemeral ML-KEM keypair (esk, epk).
   - Sends epk to Responder (Bob).
2. **Responder (Bob)**:
   - Encapsulates shared secret ss against Alice's static identity key (Static_PK_Alice).
   - Encapsulates ss against Alice's ephemeral epk.
   - Returns both ciphertexts.
3. **Session Key:** Both derive the symmetric encryption key (ChaCha20-Poly1305) from ss.

This ensures forward secrecy and mutual authentication without transmitting long-term private keys.

7.2 DENIAL OF SERVICE (DOS) MITIGATION

PQC verification is computationally heavier than ECC. To prevent compute-exhaustion attacks:

- **Pre-Handshake Puzzle:** Nodes must solve a small, memory-hard client puzzle (e.g., Equihash-lite) before the receiver attempts an ML-KEM decapsulation.
- **GossipSub Scoring:** Peers sending invalid PQC signatures are immediately scored -100 and disconnected.

7.3 FORMALIZING COMPUTATIONAL ASYMMETRY

To mitigate the risk of resource exhaustion attacks on PQC verification, Dytallix imposes a Computational Asymmetry constraint. Let W_{verify} be the CPU cycles required for a node to verify a handshake, and W_{solve} be the cycles required for an initiator to solve the client puzzle. We define a security parameter k such that:

$$W_{solve} \geq k \cdot W_{verify}$$

Where k is dynamically adjusted based on the current peer table saturation $\rho \in [0,1]$.

$$k(\rho) = \begin{cases} 0 & \text{if } \rho < 0.5 \text{ (Low Load)} \\ 100 \cdot e^{5(\rho - 0.5)} & \text{if } \rho \geq 0.5 \text{ (High Load)} \end{cases}$$

This ensures that during a DoS attempt (where $\rho \to 1$), the cost to the attacker rises exponentially relative to the verifier, effectively dampening the attack vector.

8. Dytallix Virtual Machine (DVM)

The DVM is a high-performance WASM (WebAssembly) runtime environment. Its primary deviation from standard EVM architectures is the Dimensional Cost Function, which decouples computational complexity from the unique storage burdens of PQC functionality.

8.1 THE DIMENSIONAL COST FUNCTION

A transaction tx in Dytallix does not consume a single "gas" unit. Instead, its cost is defined as a scalar product of its resource consumption vector $\mathbf{v}_{tx}$ and the network's dynamic price vector $\mathbf{p}_{net}$

Let the total fee F_{tx} be defined as:

$$F_{tx} = C_{gas} \cdot P_c + B_{gas} \cdot P_b(\mathcal{U})$$

Where:

- C_{gas} is the count of WASM instructions (CPU cycles).
- P_c is the base price for compute (relatively static).
- B_{gas} is the bandwidth consumption in bytes (Signature + CallData).
- $P_b(\mathcal{U})$ is the dynamic base price for bandwidth, dependent on block utilization $\mathcal{U}$.

8.2 EXPONENTIAL RAMPING (SPAM DEFENSE)

To prevent "Signature Spam" (bloating the state with 2.4KB+ lattice signatures), the bandwidth price P_b follows an exponential curve relative to the target block size utilization $\mathcal{U}_{target}$:
Where α is the elasticity coefficient determined by governance. This ensures that while honest

$$P_b(\mathcal{U}) = P_{base} \cdot e^{\alpha(\mathcal{U}-\mathcal{U}_{target})}$$

PQC usage is affordable, any attempt to flood the network with large signatures causes the B_{gas} price to grow exponentially and become economically prohibitive, providing an algorithmic firewall against state exhaustion.

8.3 PRECOMPILED CONTRACTS (0XADDRESS SPACE)

To enable efficient signature verification within smart contracts (e.g., for Account Abstraction wallets), the DVM exposes native Rust implementations via host functions:

- 0x10: verify_mldsa65(pubkey_ptr, msg_ptr, sig_ptr) -> bool
- 0x11: verify_slhdsa_shake(pubkey_ptr, msg_ptr, sig_ptr) -> bool
- 0x12: kem_decapsulate(privkey_ptr, cipher_ptr) -> shared_secret

These precompiles bypass the WASM overhead, calling directly into the optimized Rust pqc-core crate.

9. Crypto-Economic Security

Dytallix employs a dual-token model to separate governance security from network utility, ensuring that the heavy cryptographic security layer is economically sustainable.

9.1 ARCHITECTURE AND ROLES

The ecosystem is powered by two distinct assets, each serving a specialized role in the security model to decouple governance rigidity from the flexible emission required for PQC incentives.

Token	Name	Role	Characteristics
DGT	Dytallix Governance Token	Security & Governance	Used for DAO voting and staking delegation. Locked in contracts; not used for spending. Non-inflationary fixed supply of 1,000,000,000 DGT.
DRT	Dytallix Reward Token	Utility & Rewards	The adaptive-emission token minted by the Emission Controller. Paid to validators, stakers, and the treasury. Burnable by holders.

9.2 THE ADAPTIVE EMISSION CONTROLLER

To maintain economic security without unchecked inflation, Dytallix utilizes an Emission Controller smart contract. This controller adjusts the issuance of DRT based on network utilization parameters (min/max rate, adjustment factor, base rate) set by the DAO.
Emission Distribution:

- **40% - Validators:** Compensation for infrastructure costs and PQC computational overhead.
- **30% - Stakers:** Yield for delegating DGT to secure the consensus.
- **30% - Treasury:** Development fund for PQC research and ecosystem grants.

9.2.1 FORMAL EMISSION LOGIC (PID CONTROL LOOP)

To ensure economic stability, the emission rate E_{t+1} for the next epoch is governed by a Discrete PID Controller.

Let L be the epoch length in blocks. The error term at epoch t is defined as $e_t = u_{target} - u_t$, where u_t is the average utilization over the epoch. The emission rate adjusts according to:

$$E_{t+1} = E_{base} + K_p e_t + K_i \sum_{j=t-W}^{t} e_j + K_d(e_t - e_{t-1})$$

Where:

- K_p (Proportional): Immediate reaction to demand shocks.
- K_i (Integral): Correction for long-term historical drift over window W.
- K_d(Derivative): Finite difference term to dampen oscillation.

9.3 STAKING MODEL

The staking mechanism is designed to be lossless regarding the governance token.

- **Delegation:** Users delegate DGT to validators. This DGT is locked, not spent. It represents the weight of the vote in consensus.
- **Reward Accrual:** Stakers earn DRT from the staker pool. They never earn DGT (which is fixed).
- **Reward Index:** Rewards are tracked via a global reward index, ensuring correct accrual even during bootstrapping phases. Claims settle the index math and update user DRT balances.

9.4 DIMENSIONAL FEE MARKET

The core Dimensional Gas Model (C-Gas + B-Gas) serves as the primary firewall against PQC-specific attacks within the dual-token economy.

- **Cost of Attacks:** If an attacker attempts to spam the network with large ML-DSA signatures, the B-Gas base fee (payable in DRT) rises exponentially.
- **Economic Security:** This structure ensures that valid, low-bandwidth financial transactions remain cheap (low C-Gas), while heavy data attacks become prohibitively expensive, protecting the network from state bloat.

10. On-Chain Governance & Agility

Dytallix avoids the "hard fork for upgrades" trap by baking Crypto-Agility directly into the protocol state. This is formalized via the Algorithm Registry, a system-level state machine that governs the validity of cryptographic primitives.

10.1 FORMAL DEFINITION: THE ALGORITHM REGISTRY

Let $\mathbb{A}$ be the set of supported algorithms. The Registry $\mathscr{R}$ is defined as a mapping from a unique Algorithm ID to a tuple containing its verification logic and lifecycle state:

$$\mathscr{R} : \text{AlgID} \rightarrow \{\mathscr{V}_{code}, S_{state}\}$$

Where $S_{state} \in \{\text{Experimental}, \text{Active}, \text{Deprecated}, \text{Revoked}\}$.

10.2 STATE TRANSITION FUNCTION

The state of an algorithm evolves according to a transition function $\delta(S_t, \Phi)$, where Φ represents the governance consensus vector

$$S_{t+1} = \begin{cases} \text{Active} & \text{if } S_t = \text{Exp} \wedge \sum_{v \in V} w_v > T_{activation} \\ \text{Deprecated} & \text{if } S_t = \text{Active} \wedge \sum_{v \in V} w_v > T_{deprecation} \\ \text{Revoked} & \text{if VerifyZK}(\pi_{break}) = \text{True} \vee \text{EmergencyVote} \end{cases}$$

- Deprecated: Governance triggers a warning period. The algorithm works, but gas costs are artificially inflated to encourage migration.
- Revoked (Emergency Circuit Breaker): If a ZK-proof π_{break} demonstrates a successful forgery (e.g., finding an SVP solution $< \beta$), the transition to revoked is instantaneous, rejecting all future transactions signed by that algorithm ID at the mempool level.

11. Security & Attack Analysis

Quantifying the systemic resilience of Dytallix requires a direct comparison between classical and quantum attack complexities. This section provides a rigorous analysis of the cryptographic hardness of our chosen primitives, mapping specific attack vectors to their corresponding quantum gate counts and NIST security levels.

11.1 BIT-SECURITY ESTIMATES

The table below details the cryptographic hardness relative to the security parameter $n = 256$. For lattice schemes, β represents the BKZ block size required to solve the underlying Shortest Vector Problem.

Attack Vector	Classical Derivation (Ops)	Classical Value	Quantum Derivation (Gates)	Quantum Value	NIST Level
Key Recovery (ML-DSA-65)	$2^{0.292\beta}$ (Core-SVP)	$> 2^{192}$	$2^{0.265\beta}$ (Quantum-Sieve)	$\approx 2^{207}$	3
Pre-image (BLAKE3-256)	2^{n} (Brute Force)	2^{256}	$2^{n/2}$ (Grover)	2^{128}	1
Collision (BLAKE3-256)	$2^{n/2}$ (Birthday)	2^{128}	$2^{n/3}$ (BHT Algorithm)	$\approx 2^{85}$ *	1

*NIST PQC Levels 1, 3, and 5 are primarily defined by the classical resources required to break AES-128, AES-192, and AES-256 respectively. Quantum gate counts are provided as supplementary evidence of resistance against Grover/Shor attacks.

11.2 SIDE-CHANNEL RESISTANCE

The @dytallix/pqc-wasm and core Rust node implementation utilize:

- **Constant-Time Execution:** Rejection sampling in ML-DSA is implemented to be isochronous to prevent timing leaks.
- **Masking:** High-order masking is applied to the decryption routines of ML-KEM to protect against Power Analysis (DPA).

12. Roadmap to Mainnet

The deployment of Dytallix follows a phased execution strategy, moving from a controlled development environment to a fully decentralized, genesis-validated mainnet. This roadmap outlines the critical milestones for testing, auditing, and launching the network.

- **Phase 1 (Devnet - Active):** Dockerized localnet with simulated latency. Implementation of Noise_IK_pqc.
- **Phase 2 (Testnet "Alkali"):** Public validator onboarding. Faucet distribution. Gas schedule calibration.
- **Phase 3 (Audit):** Formal verification of the consensus-core crate and third-party audit of the WASM precompiles.
- **Phase 4 (Genesis):** Generation of the Genesis Block. Protocol-enforced invalidation of all non-PQC transactions.

13. Call for Contributors

The architecture described in this paper is not theoretical; it is running code. However, the transition to a quantum-secure web requires more than just a protocol: it requires an ecosystem of builders, cryptographers, and systems engineers willing to test the limits of this new stack.

We invite the technical community to move from observation to participation.

- **Get the SDK:** Install the @dytallix/sdk via NPM or Cargo to start generating ML-DSA signatures and building PQC-native dApps today.
- **Run a Node:** Join the Devnet by pulling the Docker container. Help us stress-test the PQC-Noise handshake under real-world network partitions.
- **Audit the Math:** Our cryptographic implementations are open-source. We challenge cryptographers to review our adaptation of Dilithium and Kyber for the blockchain context.

Visit **www.dytallix.com** to access the developer documentation, join the technical Telegram, and begin building the infrastructure that will survive the next era of computing.

14. Conclusion

The transition to Post-Quantum Cryptography is not merely a software upgrade; it is a fundamental reconstruction of the digital trust layer. As identified by BIS and NSA directives, the "Harvest Now, Decrypt Later" threat vector has already begun, rendering the timeline for PQC adoption immediate rather than distant.

Dytallix responds to this imperative with a strictly PQC-native architecture. By rejecting the "hybrid" half-measures that leave historical data exposed, Dytallix ensures:

1. **Confidentiality:** Through ML-KEM-768 transport security.
2. **Integrity:** Through ML-DSA-65 consensus and transaction signing.
3. **Longevity:** Through protocol-level Crypto-Agility and Dimensional Gas pricing.

Dytallix is not just a blockchain for the quantum era; it is the infrastructure that will allow decentralized value to survive it.