QubitChain.io: Quantum-Safe Whitepaper

Quantum-Native Blockchain Infrastructure

The world's first Layer 1 blockchain built natively on all four NIST post-quantum standards with hardware QRNG and Proof of Quantum Entropy.

Document Summary

Field	Details
Document Title	QubitChain.io Technical Whitepaper v2.0
Published	July 2026
Organization	QubitChain Research
Mainnet Target	Q1 2028
Document Type	Technical Whitepaper
Citation	QubitChain Research "QubitChain.io Quantum-Native Blockchain Infrastructure." v2.0. July 2026. qubitchain.io/whitepaper
Primary Subject	Quantum-safe blockchain architecture implementing NIST FIPS 203, FIPS 204, and FIPS 205 from genesis block.
Secondary Subjects	QRNG entropy, Proof of Quantum Entropy consensus, cryptographic agility, tokenomics, governance, NSA CNSA 2.0 compliance.

Key Technical Claims	First blockchain natively implementing all three NIST 2024 PQC standards; hardware QRNG at key generation and consensus level; no hard fork required for future algorithm transitions.

1. Abstract

The advent of fault-tolerant quantum computing represents the most significant and schedulable threat to the cryptographic foundations of blockchain infrastructure in the history of the field. Over $3.2 trillion in digital assets are currently secured by RSA, Elliptic Curve Cryptography (ECC), and ECDSA, algorithms that are provably broken by Shor's algorithm running on a Cryptographically Relevant Quantum Computer (CRQC). The Harvest Now, Decrypt Later (HNDL) threat vector means this risk is not future-dated: adversaries with archival capability and strategic patience are already harvesting public blockchain data for retrospective decryption.

In August 2024, NIST published three finalized post-quantum cryptographic standards: FIPS 203 (ML-KEM), FIPS 204 (ML-DSA), and FIPS 205 (SLH-DSA). In 2025, NIST added a fourth standard, HQC, providing code-based cryptographic diversity alongside the lattice-based primary algorithms. These standards represent the global regulatory baseline for quantum-safe digital security, mandated for U.S. national security systems by NSA CNSA 2.0 and for high-risk sector operators in the EU by the EU PQC Roadmap.

QubitChain.io introduces a natively quantum-safe Layer 1 blockchain built on all four NIST post-quantum standards from its genesis block, with hardware Quantum Random Number Generator (QRNG) entropy at both key generation and consensus randomness levels, and a novel consensus mechanism (Proof of Quantum Entropy, PoQE) that cannot be predicted or manipulated by any adversary regardless of computational capability. This post-quantum blockchain whitepaper provides the complete technical, economic, and governance specification for the QubitChain.io protocol.

2. Executive Summary

QubitChain.io is a Layer 1 blockchain protocol with the following defining properties:

- **Cryptographic Foundation:** All transaction signatures use ML-DSA (FIPS 204). All key encapsulation uses ML-KEM (FIPS 203). SLH-DSA (FIPS 205) is available as a backup signature scheme with independent security assumptions. HQC (NIST 2025) is incorporated via the cryptographic agility framework.
- **Key Generation Security:** All private keys are generated using entropy from certified hardware QRNG devices, eliminating the classical PRNG entropy attack surface entirely.
- **Consensus Mechanism:** Proof of Quantum Entropy (PoQE) selects validators using

verifiable entropy from hardware QRNG sources committed on-chain before selection. The selection cannot be predicted or manipulated by any adversary, classical or quantum.
- **Cryptographic Agility:** The modular cryptographic layer allows algorithm transitions via governance-approved protocol parameter updates with no hard fork required.
- **Regulatory Compliance:** QubitChain.io satisfies NSA CNSA 2.0 requirements, NIST FIPS 203/204/205 compliance requirements, and EU PQC Roadmap requirements from its genesis block.
- **Network Architecture:** Full-stack post-quantum security including P2P layer ML-KEM encryption, quantum-safe smart contract execution, and a cross-chain bridge for migration from classical blockchain networks.

Mainnet Target: Q1 2028, following public testnet alpha in Q1 to Q2 2027.

3. The Quantum Threat to Blockchain Infrastructure

3.1 Shor's Algorithm and the Collapse of Elliptic Curve Security

[Shor's Algorithm]: A quantum algorithm solving integer factorization and discrete logarithm problems in polynomial time, which breaks RSA, ECDSA, and Ed25519 on a CRQC.

Peter Shor's 1994 algorithm demonstrates that a quantum computer with sufficient logical qubits can solve both the integer factorization problem (underlying RSA security) and the discrete logarithm problem (underlying ECDSA and EdDSA security) in polynomial time. The specific impact on blockchain networks:

- **ECDSA on secp256k1 (Bitcoin, Ethereum, Avalanche):** A CRQC running Shor's algorithm can derive the private key from any exposed public key in polynomial time. The elliptic curve discrete logarithm problem provides no quantum resistance.
- **EdDSA on Ed25519 (Solana, Cardano, Polkadot):** The discrete logarithm problem on the Edwards25519 curve is equally broken by Shor's algorithm. Ed25519's efficiency and clean implementation do not affect its quantum vulnerability.
- **RSA (used in TLS, PKI, and some blockchain identity systems):** The integer factorization problem underlying RSA is solved by Shor's algorithm for key sizes up to RSA-2048 with an estimated 4,000 error-corrected logical qubits.

Research published by Webber et al. in *AVS Quantum Science* (2022) estimates that breaking Bitcoin's ECDSA-256 encryption requires approximately 4 million physical qubits at current error rates, or approximately 317 fully fault-tolerant logical qubits. Multiple independent quantum hardware programs are tracking toward this threshold on timelines consistent with NSA and CISA public threat assessments.

3.2 Grover's Algorithm and Hash Function Security

Grover's algorithm provides a quadratic speedup for unstructured search problems, effectively halving the security level of symmetric key algorithms and hash functions. The specific impact:

- **SHA-256 (Bitcoin mining):** Effective security reduced from 256-bit to 128-bit. This is a

significant but non-catastrophic degradation in the near term.
- **Keccak-256 (Ethereum):** Same analysis as SHA-256 under Grover's speedup.
- **SHA-512 and BLAKE3 (QubitChain.io):** At 256-bit output, these provide 256-bit post-quantum security under Grover's algorithm, maintaining full security margins.

The combined effect of Shor's attack on ECDSA signatures and Grover's degradation of SHA-256 hashing means the complete trust model of classical blockchain infrastructure collapses under quantum attack, not merely one component of it.

3.3 The Harvest Now, Decrypt Later (HNDL) Threat

[HNDL]: An attack strategy where an adversary archives encrypted or blockchain data today for future decryption once a CRQC exists.

The HNDL attack vector fundamentally changes the timeline of the quantum threat from future to present. Because all Bitcoin and Ethereum transaction data is publicly available and permanently recorded on-chain, adversaries with storage capacity can archive the complete UTXO set, all exposed public keys, and all transaction history at negligible cost and zero ongoing access requirements.

The critical implications:

- An estimated 25 percent of all circulating Bitcoin supply is in Pay-to-Public-Key (P2PK) outputs or reused Pay-to-Public-Key-Hash (P2PKH) outputs where public keys are permanently and irremovably on-chain. This includes coins widely attributed to Satoshi Nakamoto.
- Every Ethereum EOA (Externally Owned Account) that has ever signed a transaction has its public key permanently on-chain.
- CISA has formally stated in published guidance that organizations should assume nation-state adversaries are already collecting data for future quantum decryption.
- The NSA's CNSA 2.0 mandate was issued specifically because classified intelligence about state-sponsored quantum programs made the urgency of the HNDL threat impossible to ignore at the national security level.

3.4 Hardware Progress: The Accelerating Timeline

[CRQC]: A quantum computer with sufficient logical qubit count, fidelity, and gate count to run Shor's algorithm at the scale required to break RSA-2048 or ECDSA-256 within a practically useful time frame.

The convergence of multiple independent quantum hardware programs across multiple hardware architectures and multiple national programs, all tracking toward CRQC capability within the same decade, makes the probability that none of them succeeds before 2035 extremely low. Key hardware programs driving this timeline include IBM's Condor and Heron R2 platforms, Google's Willow chip, Microsoft's Majorana 1 topoconductor architecture, and rapidly scaling trapped-ion systems from IonQ and Quantinuum.

4. NIST Post-Quantum Cryptographic Standards

What NIST standards does QubitChain implement?

- **FIPS 203 ML-KEM:** Module-Lattice-Based Key-Encapsulation Mechanism for P2P encryption.
- **FIPS 204 ML-DSA:** Module-Lattice-Based Digital Signature Algorithm for transaction signing.
- **FIPS 205 SLH-DSA:** Stateless Hash-Based Digital Signature Algorithm as backup.
- **HQC 2025:** Hamming Quasi-Cyclic code-based key encapsulation via cryptographic agility.

QubitChain.io implements all four NIST post-quantum cryptographic standards as native protocol primitives.

FIPS 203: ML-KEM (Module Lattice Key Encapsulation Mechanism)

[ML-KEM]: NIST FIPS 203. The post-quantum standard for key encapsulation, based on MLWE. Used by QubitChain.io for all key encapsulation operations.

Standardized from CRYSTALS-Kyber. Used for all key encapsulation operations across the QubitChain.io network, including node-to-node P2P session key establishment, user wallet key exchange, and validator communication encryption. Based on the Module Learning With Errors (MLWE) problem in structured lattices. No known quantum algorithm provides an efficient speedup for MLWE.

QubitChain.io implements ML-KEM-768 (NIST security level 3, comparable to AES-192) as the network standard, with ML-KEM-1024 (level 5, comparable to AES-256) available for high-value or extended-lifetime operations.

FIPS 204: ML-DSA (Module Lattice Digital Signature Algorithm)

[ML-DSA]: NIST FIPS 204. Standardized from CRYSTALS-Dilithium, which replaces ECDSA as a lattice-based post-quantum signature scheme with no known quantum attack path. It provides robust security for blockchain transactions against both classical and quantum adversaries.

The primary digital signature scheme for all QubitChain.io transaction authorization, validator attestation, block signing, and governance vote signing. Based on the same MLWE mathematical structure as ML-KEM.

QubitChain.io implements ML-DSA-65 (level 3) as the default transaction signature scheme, providing 2,420-byte signatures with strong quantum security margins. ML-DSA-87 (level 5) is available for validator key signing operations and high-value institutional transactions.

What is the difference between ML-DSA and ECDSA?

FEATURE	ML-DSA (QUBITCHAIN.IO)	ECDSA (BITCOIN/ETHEREUM)
Standard	NIST FIPS 204 post-quantum standard	Classical digital signature standard
Security Basis	Module Learning With Errors (Lattice)	Elliptic Curve Discrete Logarithm
Quantum Security	Secure (No known quantum attack)	Broken by Shor's algorithm on a CRQC
Signature Size	Large (2,420 bytes for ML-DSA-65)	Small (71 bytes for secp256k1)

QubitChain.io uses ML-DSA for transaction signing. Bitcoin and Ethereum use ECDSA. ECDSA is broken by Shor's algorithm. ML-DSA is not.

FIPS 205: SLH-DSA (Stateless Hash-Based Digital Signature Algorithm)

[SLH-DSA]: NIST FIPS 205. A post-quantum signature scheme based on hash functions with no algebraic structure, providing independent security assumptions from ML-DSA.

Standardized from SPHINCS+. A stateless hash-based signature scheme whose security rests entirely on the security of cryptographic hash functions, with no algebraic structure of any kind. This provides cryptographic diversity: SLH-DSA and ML-DSA have completely independent security assumptions. A theoretical breakthrough against lattice-based cryptography would not affect SLH-DSA security.

HQC (NIST 2025 Standard): Code-Based Key Encapsulation

[HQC]: A code-based post-quantum key encapsulation mechanism standardized by NIST in 2025, providing a mathematically independent alternative to lattice-based ML-KEM.

HQC (Hamming Quasi-Cyclic) was standardized by NIST in 2025 as a fourth post-quantum standard. It is a code-based key encapsulation mechanism whose security rests on the decoding problem for random quasi-cyclic codes, a third distinct mathematical foundation from both lattice-based (ML-KEM, ML-DSA) and hash-based (SLH-DSA) schemes. QubitChain.io incorporates HQC adoption as a planned activation within the cryptographic agility framework. No hard fork is required for HQC activation.

4.2 Mathematical Security Basis

[MLWE]: The mathematical hardness problem underlying ML-KEM and ML-DSA. No known classical or quantum algorithm solves MLWE efficiently.

The security of ML-KEM and ML-DSA rests on the Module Learning With Errors (MLWE) problem. The security of SLH-DSA rests on the security of the underlying hash function under Grover's algorithm. The security of HQC rests on the decoding problem for random linear codes. The combination of ML-KEM/ML-DSA (lattice), SLH-DSA (hash), and HQC (code) provides three mathematically independent security foundations. A catastrophic breakthrough against any one of these would not affect the others.

5. QRNG Architecture

[QRNG]: Quantum Random Number Generators are hardware devices that derive randomness from inherently non-deterministic physical quantum processes, producing genuinely non-deterministic outputs that cannot be reproduced or predicted under any computational model. This provides true physical entropy for secure cryptographic key generation on the blockchain.

5.1 The Problem with Classical Entropy

Classical pseudorandom number generators, even those using hardware entropy sources, produce outputs that are algorithmically deterministic given knowledge of the seed and internal state. This creates an attack surface for private key compromise that is independent of the signature algorithm used. Historical examples include the 2012 weak RSA keys discovery, the NSA Dual EC DRBG backdoor, and cryptocurrency wallet compromises where weak randomness sources produced predictable keys.

5.2 QRNG Architecture

QubitChain.io uses certified hardware Quantum Random Number Generators that derive entropy from inherently non-deterministic physical quantum processes, including Photon Detection Timing, Vacuum Fluctuations, and Radioactive Decay Timing. QRNG outputs are post-processed through a NIST SP 800-90B-compliant health testing framework to verify statistical quality and detect hardware failures before entropy is used in cryptographic operations.

5.3 Distributed QRNG Entropy Pool

The QubitChain.io network operates a distributed QRNG entropy pool rather than relying on a single centralized entropy source. Validator nodes contribute QRNG-sourced entropy to a shared pool using a verifiable secret sharing scheme that ensures no single node controls the entropy used for network-wide operations. Entropy contributions are committed to the chain before they are combined, allowing any network participant to verify the integrity of the entropy pool.

6. Cryptographic Agility Framework

[Cryptographic Agility]: The ability to transition between cryptographic algorithms at the protocol level without requiring a hard fork or breaking backward compatibility. QubitChain.io implements this through a modular cryptographic engine and a governance-controlled algorithm registry to seamlessly absorb future NIST post-quantum standards.

6.1 Why Cryptographic Agility Is Required

No cryptographic algorithm is permanently secure. A blockchain without cryptographic agility faces two options when an algorithm needs to change: live with the weakening algorithm indefinitely, or execute a contested community hard fork. For a network securing significant assets, neither option is acceptable.

6.2 QubitChain.io's Agility Architecture

QubitChain.io's cryptographic layer is abstracted from the protocol logic. Transaction and block formats include an explicit algorithm identifier field, allowing the network to recognize, validate, and produce signatures using any algorithm registered in the cryptographic registry. The algorithm registry is a governance-controlled protocol parameter.

6.3 Backward Compatibility During Transitions

During an algorithm transition period, the network accepts signatures from both the outgoing and incoming algorithm. This allows existing wallets using the outgoing algorithm to remain valid while users migrate to the new default. The transition timeline is enforced by protocol parameter update.

7. Proof of Quantum Entropy Consensus

Proof of Quantum Entropy (PoQE) is QubitChain.io's consensus mechanism where validator selection uses verifiable entropy from hardware QRNG sources committed on-chain before selection. The selection randomness cannot be predicted or manipulated by any adversary, classical or quantum, because it depends entirely on inherently non-deterministic physical quantum processes rather than computable functions.

7.1 Motivation: Why Classical Consensus Randomness Is Insufficient

All existing Proof of Stake consensus mechanisms use some form of classical randomness for validator selection (RANDAO, Ouroboros Praos VRF, Solana VDF). These are ultimately based on classical cryptographic primitives. An adversary with sufficient classical or quantum computational resources can analyze the entropy inputs and attempt to predict or bias validator selection. The correct solution is to source randomness from a physical quantum random source.

7.2 PoQE Mechanism Specification

The QubitChain.io network operates in epochs of fixed block count. During the commitment phase (the final blocks of each epoch), each active validator commits a hash commitment of their QRNG-sourced entropy contribution. In the first block of each new epoch, validators reveal their committed entropy values. Revealed entropy values are combined using XOR followed by SHA-512 to produce the epoch randomness beacon, which seeds a deterministic selection algorithm for the new epoch.

7.3 PoQE vs Classical Consensus: Security Comparison

PROPERTY	CLASSICAL POS (VRF)	POQE (QUBITCHAIN.IO)
Randomness Source	Algorithmic VRF (deterministic)	Hardware QRNG (physically non-deterministic)
Prediction by Quantum Adversary	Possible (VRF based on ECC/hash)	Impossible (quantum processes cannot be computed)
Manipulation Resistance	Requires computational advantage	Requires physical hardware control
Verifiability	Cryptographic proofs verifiable classically (VRF)	Commitment scheme verifiable by any observer
Energy Consumption	Low	Low (equivalent to classical PoS)

7.4 PoQE and Sybil Resistance

The PoQE consensus mechanism incorporates QRNG-backed identity verification: validators must provide cryptographic proof that their identity key was generated from a certified QRNG source. This makes identity multiplication (Sybil attacks) structurally harder, because each additional identity requires an additional QRNG hardware commitment rather than simply additional computational resources.

8. Network Architecture and Protocol Specification

8.1 Transaction Structure

Every QubitChain.io transaction contains protocol version, algorithm identifier, sender address (a hash commitment derived from ML-DSA public key), recipient address, value, nonce, optional smart contract calldata, and an ML-DSA signature over the transaction hash. For

validator transactions, an additional QRNG entropy commitment is included. The public key is NOT embedded in the address by default.

8.2 Block Structure

Block Headers include protocol version, previous block hash, timestamp, epoch number, slot number, proposer identity, epoch randomness beacon, state root, transaction root, and QRNG entropy commitment. Block Bodies include ordered transactions, attestation aggregates, and validator signature on the header using ML-DSA.

8.3 P2P Network Layer

All node-to-node communication is encrypted using ML-KEM session keys. The handshake protocol establishes a shared secret using ML-KEM-768 key encapsulation before any network message content is transmitted. Node identity is established through ML-DSA key pairs, and all inter-node messages are authenticated using ML-DSA signatures to prevent man-in-the-middle attacks.

8.4 Address Derivation

Addresses are derived by generating a private key using QRNG entropy, deriving the ML-DSA public key, computing the public key hash using SHA-512 and BLAKE3, and encoding the 32-byte hash output using base58check. Revealing the public key in the signing transaction does not create quantum vulnerability because the signature scheme is ML-DSA (not ECDSA).

9. Smart Contract Layer

QubitChain.io's smart contract layer is designed as a quantum-safe execution environment where all interactions between users and contracts, and between contracts, use ML-DSA signatures for authorization. Smart contracts deployed on QubitChain.io inherit quantum-safe security at the protocol level.

Contract function calls are authorized by ML-DSA signatures from the calling address. Contracts cannot generate private keys internally (preventing reentrancy-style entropy injection). Key generation requires an explicit QRNG entropy request.

QubitChain.io publishes native token standards for fungible tokens (QRC-20 equivalent) and non-fungible tokens (QRC-721 equivalent) that incorporate quantum-safe authorization as a required field.

10. Cross-Chain Bridge

The cross-chain bridge enables secure asset migration from classical blockchain networks (Bitcoin, Ethereum, and others) to QubitChain.io. This addresses the practical need for users who hold assets on quantum-vulnerable networks to move to quantum-safe infrastructure

before Q-Day.

The bridge operates through a network of ML-DSA-secured relay nodes that monitor classical networks for deposit transactions and mint wrapped assets on QubitChain.io via threshold ML-DSA signatures. Relay nodes are selected using the PoQE mechanism, ensuring that selection cannot be manipulated by adversaries targeting the bridge's relay network.

11. Tokenomics and Economic Model

The QubitChain.io native token serves three functions: Transaction Fees, Validator Staking, and Governance. Fee mechanisms follow a base fee plus priority fee model analogous to EIP-1559 on Ethereum, with base fees burned and priority fees paid to block proposers.

The fee structure accounts for the larger transaction sizes inherent in ML-DSA signatures relative to ECDSA. Network throughput targets are set with ML-DSA signature sizes (2,420 bytes for ML-DSA-65) as the baseline assumption, ensuring throughput is engineered correctly for the actual protocol.

12. Governance Framework

QubitChain.io's on-chain governance covers the cryptographic algorithm registry, PoQE consensus parameters, bridge relay node requirements, validator requirements, smart contract upgrades, and emergency security response parameters.

Proposals pass with a two-thirds supermajority of participating stake and a minimum quorum of 33 percent of total staked supply. Algorithm additions receive the highest governance scrutiny, requiring a formal cryptographic assessment submitted with the proposal and an extended discussion period.

13. Validator Requirements and Staking

Validators must meet strict hardware requirements to ensure network integrity, including a physically integrated certified hardware QRNG device (NIST SP 800-90B), adequate compute hardware for ML-DSA verification, network bandwidth, and geographic diversity.

Validators bond native tokens as economic collateral and are subject to slashing conditions such as equivocation, missing commitments, or QRNG manipulation attempts. Native token holders who do not operate validator nodes can delegate stake to active validators.

14. NSA CNSA 2.0 Compliance

QubitChain.io's native implementation of ML-KEM and ML-DSA (the two primary algorithms mandated by CNSA 2.0 for all U.S. national security systems) means the network is fully compliant with CNSA 2.0 requirements from its genesis block.

CNSA 2.0 deprecates ECDSA and RSA; QubitChain.io never uses either. Review the NSA CNSA

2.0 advisory for explicit guidance.

QubitChain.io implements FIPS 203 (ML-KEM) for key encapsulation, FIPS 204 (ML-DSA) for digital signatures, FIPS 205 (SLH-DSA) as a backup, and HQC via cryptographic agility. This satisfies the EU PQC Roadmap and UK NCSC Quantum Readiness guidelines natively.

15. Security Threat Model

- **Nation-State with QRNG Control:** Mitigated via geographic distribution requirements for validators and distributed entropy pool aggregation.
- **Nation-State with CRQC:** Mitigated via ML-KEM and ML-DSA, which have no known quantum attack path, with SLH-DSA backup and cryptographic agility.
- **51 Percent Stake Attack:** Mitigated because PoQE's QRNG-based selection means stake weight influences but does not strictly determine selection.
- **Cross-Chain Bridge Attack:** Mitigated through threshold signatures, PoQE relay selection, and ML-KEM/ML-DSA encryption.

16. Performance Specifications

Metric	Specification
Target Transaction Throughput	1,000 TPS at mainnet launch, scaling roadmap to 10,000 TPS through parallel execution
Target Block Time	2 to 4 seconds.
Target Finality	Under 10 seconds (single epoch finality)
Standard Transfer Size	Approximately 2,600 bytes, dominated by the ML-DSA-65 signature
QRNG Key Generation	Under 100ms per key pair on reference hardware
ML-DSA-65 Verification	Under 2ms on reference hardware

17. Comparison with Existing Post-Quantum Approaches

QubitChain.io uses ML-DSA for transaction signing. Bitcoin and Ethereum use ECDSA. ECDSA is broken by Shor's algorithm. ML-DSA is not.

17.1 QubitChain.io vs QRL (Quantum Resistant Ledger)

QRL launched in 2018 using XMSS, a stateful hash-based signature scheme. QubitChain.io uses stateless ML-DSA, implements all four NIST standards, uses PoQE with QRNG instead of Proof of Work, uses ML-KEM for P2P encryption, and provides a cryptographic agility framework.

17.2 QubitChain.io vs Ethereum PQC Migration Plans

Ethereum's PQC migration research requires a community-wide hard fork and multi-year migration. QubitChain.io implements ML-KEM and ML-DSA natively from genesis with no migration dependency.

17.3 QubitChain.io vs Algorand State Proofs

Algorand demonstrated Falcon signatures via State Proofs, but main transactions still use Ed25519. QubitChain.io applies ML-DSA to all transaction authorization from genesis.

18. Development Roadmap

QubitChain.io mainnet is targeted for Q1 2028, following public testnet alpha in Q1 to Q2 2027.

- **Phase 1: Cryptographic Foundation (Q3 2025 to Q2 2026, Completed)**
  FIPS 203, 204, 205 protocol integration. QRNG hardware specification. PoQE theoretical model validated. HQC integration architecture designed.
- **Phase 2: Network Architecture (Q3 2026 to Q4 2026)**
  P2P layer ML-KEM encryption. QRNG entropy pool architecture. Closed internal testnet launch. Developer documentation alpha.
- **Phase 3: Public Testnet and Developer Access (Q1 2027 to Q3 2027)**
  Public testnet alpha. Developer SDK release. Smart contract environment beta. Validator onboarding. Bug bounty program.
- **Phase 4: Security Audit and Hardening (Q3 2027 to Q4 2027)**
  Third-party security audits of protocol, bridge, and QRNG integration. Mainnet genesis block parameters finalized.
- **Phase 5: Mainnet Launch (Q1 2028)**
  Mainnet genesis block. Validator network live. Smart contract platform open. Cross-chain bridge operational.
- **Phase 6: Ecosystem and Scale (Q2 2028 onwards)**
  Throughput scaling upgrades. DeFi protocol launches. Custody integrations. Government and defense partnership deployments.

19. Risk Factors

- **Quantum Hardware Timeline Uncertainty:** If quantum hardware advances faster than projected, existing networks may experience security failures before QubitChain.io reaches mainnet.
- **Algorithm Vulnerability Risk:** A theoretical breakthrough against ML-DSA or ML-KEM

would require immediate exercise of the cryptographic agility mechanism.
- **Mainnet Launch Timeline Risk:** Security audit findings requiring significant remediation could delay the Q1 2028 mainnet target.
- **Adoption and Regulatory Risks** are also factored into the network's strategic planning.

20. References

- NIST FIPS 203: Module-Lattice-Based Key-Encapsulation Mechanism Standard (August 2024). csrc.nist.gov
- NIST FIPS 204: Module-Lattice-Based Digital Signature Standard (August 2024). csrc.nist.gov
- NIST FIPS 205: Stateless Hash-Based Digital Signature Standard (August 2024). csrc.nist.gov
- NSA, "Commercial National Security Algorithm Suite 2.0 Cybersecurity Advisory". media.defense.gov
- Shor, P.W. (1994). "Algorithms for Quantum Computation: Discrete Logarithms and Factoring."
- Grover, L.K. (1996). "A Fast Quantum Mechanical Algorithm for Database Search."
- Webber et al. (2022). "The Impact of Hardware Specifications on Reaching Quantum Advantage in the Fault Tolerant Regime." *AVS Quantum Science* 4(1).
- Boudot et al. (2020). "Comparing Elliptic Curve Discrete Logarithm Algorithms on GPUs." *IACR ePrint*.
- Google Quantum AI (2024). "Quantum Error Correction Below the Surface Code Threshold." *Nature*.
- IBM Quantum Roadmap (2024).
- Bernstein and Lange (2022). "CRYSTALS-Dilithium: Algorithm Specifications."

21. Glossary

[CRQC]: A quantum computer with sufficient logical qubit count, fidelity, and gate count to run Shor's algorithm at the scale required to break RSA-2048 or ECDSA-256 within a practically useful time frame.

[Cryptographic Agility]: The ability to transition between cryptographic algorithms at the protocol level without requiring a hard fork or breaking backward compatibility.

[ECDSA]: Elliptic Curve Digital Signature Algorithm: The signature scheme used by Bitcoin, Ethereum, Avalanche, and most major blockchain networks, which is broken by Shor's algorithm on a CRQC.

[Grover's Algorithm]: A quantum algorithm providing quadratic speedup for unstructured search, halving the effective security level of hash functions.

[HNDL]: Harvest Now, Decrypt Later: An attack strategy where an adversary archives encrypted or blockchain data today for future decryption once a CRQC exists.

[ML-DSA]: NIST FIPS 204. The post-quantum standard for digital signatures, based on MLWE. Used by QubitChain.io for all transaction and validator signatures.

[ML-KEM]: NIST FIPS 203. The post-quantum standard for key encapsulation, based on MLWE. Used by QubitChain.io for all key encapsulation operations.

[PoQE]: Proof of Quantum Entropy: QubitChain.io's consensus mechanism where validator selection uses verifiable entropy from hardware QRNG sources committed on-chain before selection.

[QRNG]: Quantum Random Number Generator: A hardware device deriving randomness from inherently non-deterministic physical quantum processes, producing non-deterministic outputs.

Appendix A: Algorithm Parameter Sets

ML-KEM Parameter Sets (FIPS 203)

PARAMETER SET	SECURITY LEVEL	PUBLIC KEY SIZE	CIPHERTEXT SIZE	SHARED SECRET
ML-KEM-512	Level 1 (AES-128)	800 bytes	768 bytes	32 bytes
ML-KEM-768	Level 3 (AES-192)	1,184 bytes	1,088 bytes	32 bytes
ML-KEM-1024	Level 5 (AES-256)	1,568 bytes	1,568 bytes	32 bytes

QubitChain.io network standard: ML-KEM-768. High-value option: ML-KEM-1024.

ML-DSA Parameter Sets (FIPS 204)

PARAMETER SET	ML-DSA-44	ML-DSA-65	ML-DSA-87
SECURITY LEVEL	Level 2	Level 3	Level 5
PUBLIC KEY	1,312 bytes	1,952 bytes	2,592 bytes
PRIVATE KEY	2,528 bytes	4,000 bytes	4,864 bytes
SIGNATURE	2,420 bytes	3,293 bytes	4,595 bytes

QubitChain.io default transaction: ML-DSA-65. Validator key signing: ML-DSA-87.

Appendix B: PoQE Entropy Commitment Scheme

Commitment Phase: For each validator i in the active set, generate QRNG entropy e_i of 512 bits. Compute commitment $c_i = \text{SHA-512}(\text{nonce}_i \parallel e_i)$. Publish c_i on-chain.

Reveal Phase: Each validator broadcasts (nonce_i, e_i). The network verifies that:

$$ \text{SHA-512}(\text{nonce}_i \parallel e_i) = c_i $$

Unmatched reveals are rejected and penalized.

Aggregation: Compute beacon:

$$ B = \text{SHA-512}(\bigoplus e_i) $$

B is the epoch randomness beacon used for validator selection.

Security Note: XOR aggregation ensures that a single validator's contribution affects the beacon output. A validator who wants to bias the beacon would need to know all other validators' QRNG entropy in advance, which is impossible.

QubitChain.io

The world's first natively quantum-resistant blockchain infrastructure. Built on NIST-standardized Post-Quantum Cryptography for the post-quantum era.

Contact: contact@qubitchain.io

Connect: LinkedIn | GitHub | Instagram | YouTube | X