Quantum Threat Model for Blockchain Infrastructure

Zach Kelling

Lux Industries

Original: 2023 | Revised: February 2026

Abstract

We present a systematic threat model for quantum computing attacks against blockchain infrastructure. Unlike prior analyses that focus narrowly on signature forgery, we enumerate the full attack surface: mempool public key exposure, validator signature forgery, retroactive chain compromise via historical transaction replay, bridge key extraction, and oracle manipulation through forged attestations. We provide timeline estimates for cryptographically relevant quantum computers (CRQCs) based on published hardware roadmaps from IBM, Google, and PsiQuantum, concluding that a >50% probability of CRQC existence before 2035 mandates immediate defensive action. We perform an economic analysis showing that the cost of hybrid post-quantum deployment (approximately 15% throughput overhead and 1.2 KB additional signature data per epoch) is negligible compared to the expected loss from a successful quantum attack on a major blockchain (total value at risk exceeding $500 billion across all chains). The rational strategy is immediate hybrid deployment, not wait-and-see. We conclude with concrete recommendations for blockchain protocol designers.

Keywords: quantum computing, blockchain security, Shor's algorithm, Grover's algorithm, post-quantum migration, threat model

Contents

1 Introduction

The security of every major blockchain rests on the hardness of the discrete logarithm problem (DLP) on elliptic curves. Bitcoin uses ECDSA on secp256k1. Ethereum uses ECDSA on secp256k1 for transactions and BLS12-381 for consensus. Lux uses BLS12-381 for consensus, ECDSA for C-Chain transactions, and EdDSA for bridge operations. Shor's algorithm [1] solves DLP in polynomial time on a quantum computer, breaking all of these schemes simultaneously.

This paper is not about whether quantum computers will eventually break blockchain cryptography—they will, given sufficient scale. It is about *when* and *how*, and what the consequences are for systems that do not prepare.

The central argument is that blockchain is *uniquely* vulnerable to quantum attack compared to other cryptographic systems, for three reasons:

1. **Permanent public record**: Every signature ever produced is stored on-chain forever. There is no concept of session expiry.
2. **Harvest-now-decrypt-later**: An adversary need not have a quantum computer today. Recording signed transactions and public keys today is sufficient to mount attacks years later.
3. **Economic persistence**: Unlike a TLS session protecting a single HTTP request, blockchain signatures protect assets that may appreciate in value over decades.

2 Shor's Algorithm: Impact on Blockchain Primitives

2.1 Algorithm Overview

Shor's algorithm [1] finds the order of an element in a finite group using quantum phase estimation. For an element g of order r in a group $\mathbb{G}$, Shor's algorithm finds r in $O((\log|\mathbb{G}|)^2 \cdot \log\log|\mathbb{G}|)$ quantum gates, with $O(\log|\mathbb{G}|)$ qubits. Given the order, the discrete logarithm follows by standard reductions.

2.2 ECDSA (secp256k1)

ECDSA security rests on the difficulty of computing k from kG where G is the generator of a 256-bit elliptic curve group. Shor's algorithm solves this with approximately 2,330 logical qubits and 1.26×10^{11} T-gates for secp256k1 [3]. With a $1\,\mu$s T-gate time and a surface code implementing error correction:

Table 1: Quantum resource estimates for breaking elliptic curve DLP.

Curve	Logical Qubits	T-gates	Est. Time
secp256k1 (256-bit)	2,330	1.26×10^{11}	8 hours
Ed25519 (255-bit)	2,310	1.24×10^{11}	8 hours
P-384 (384-bit)	3,484	4.16×10^{11}	26 hours

Resource estimates from Roetteler et al. [3], assuming surface code with code distance 27 and $1\,\mu$s cycle time.

The critical observation: once a quantum computer exists with sufficient qubits, key recovery is measured in *hours*, not years. There is no gradual degradation.

2.3 BLS12-381

BLS signatures operate on the pairing-friendly curve BLS12-381, where security rests on the co-CDH assumption in groups $\mathbb{G}_1$ (381-bit) and $\mathbb{G}_2$ (762-bit). Shor's algorithm applies to both groups. Breaking BLS12-381 requires approximately 3,800 logical qubits—more than secp256k1 due to the larger field, but the same order of magnitude.

BLS is more dangerous to break than ECDSA in the blockchain context because:

- Validator BLS keys are *static*: they persist for the validator's entire lifetime, often years.
- BLS aggregate signatures compress n validator votes into 48 bytes. If BLS is broken, an attacker can forge consensus certificates for any historical or future block.
- A forged BLS certificate can rewrite chain history or create conflicting finalized chains—a safety violation, not merely a liveness issue.

2.4 RSA

While RSA is not commonly used in blockchain consensus, it appears in:

- TLS certificates for RPC endpoints
- Verifiable delay functions (VDFs) based on RSA groups
- Some oracle attestation schemes

Shor's algorithm factors RSA-2048 with approximately 4,098 logical qubits. RSA-3072 requires approximately 6,146 logical qubits. Both are within the projected capability of CRQCs by 2035.

3 Grover's Algorithm: Hash Function Impact

Grover's algorithm [2] provides a quadratic speedup for unstructured search: finding a preimage of an n-bit hash requires $O(2^{n/2})$ quantum evaluations instead of $O(2^n)$ classical evaluations.

Table 2: Impact of Grover's algorithm on hash functions used in blockchain.

Hash	**Classical Security**	**Quantum Security**	**Status**
SHA-256	2^{256} (preimage)	2^{128}	Safe
SHA-256	2^{128} (collision)	2^{85}	Marginal
Keccak-256	2^{256} (preimage)	2^{128}	Safe
Keccak-256	2^{128} (collision)	2^{85}	Marginal
BLAKE2b-256	2^{256} (preimage)	2^{128}	Safe
Poseidon	2^{128} (algebraic)	2^{85}	Review needed

The collision resistance reduction to 2^{85} is concerning but not immediately critical: 2^{85} quantum hash evaluations still requires substantial quantum resources. However, Merkle tree constructions that rely on collision resistance (state tries, transaction trees) have reduced security margins. For chains using 128-bit collision resistance as a security parameter, upgrading to 256-bit output hashes (already standard in SHA-256 and Keccak-256) provides adequate quantum margins.

3.1 Impact on Proof-of-Work

For proof-of-work chains, Grover's algorithm provides a quadratic mining speedup. A quantum miner with 2^{40} quantum evaluations per second would match a classical miner with 2^{80} hashes per second. This is a competitive advantage, not a security break: difficulty adjustment compensates. However, a quantum miner could out-compete all classical miners combined if the quantum hardware gap is sufficiently large, enabling 51% attacks. Proof-of-stake chains (including Lux) are not affected by this vector.

4 Blockchain-Specific Attack Vectors

4.1 Attack 1: Mempool Key Exposure

When a user broadcasts a transaction, the transaction includes a signature from which the public key can be recovered (ECDSA recovery) or is explicitly included. Between broadcast and confirmation (typically 2–30 seconds on modern chains), the public key is visible in the mempool. A quantum adversary monitoring the mempool can:

1. Observe the public key in the pending transaction.
2. Compute the private key via Shor's algorithm.
3. Broadcast a competing transaction draining the account with higher gas.
4. The competing transaction confirms first (front-running).

This attack requires completing Shor's algorithm within the block time. Current estimates suggest this is infeasible with near-term quantum computers (hours per key recovery), but may become feasible as quantum hardware improves.

Mitigation: Commit-reveal transaction schemes, where the transaction is committed (hash only) before revealing the public key. However, this doubles latency and is not backward compatible with existing EVM transaction formats.

Better mitigation: Post-quantum signatures. If the signature scheme is quantum-resistant, public key exposure in the mempool is not a threat.

4.2 Attack 2: Validator Signature Forgery

Validator BLS keys are published in the validator set and remain static for the validator's entire stake duration. A quantum adversary can:

1. Recover any validator's BLS private key from their published public key.
2. Forge signatures for arbitrary blocks, including blocks that conflict with the canonical chain.
3. Create a forged finality certificate (aggregate signature) for a conflicting chain, violating the safety property of the consensus protocol.

This is the most severe quantum attack on proof-of-stake blockchains: it breaks *finality*, the property that confirmed transactions cannot be reversed.

4.3 Attack 3: Retroactive Chain Compromise

The permanent, public nature of blockchain data enables retroactive attacks:

1. A quantum adversary recovers private keys from historical public keys recorded on-chain.
2. Transfers all assets from accounts that have ever had their public key exposed (any account that has sent a transaction).
3. For UTXO chains: any output sent to a P2PKH address (public key hash) is safe until spent, but any output sent to a P2PK address (raw public key) is immediately vulnerable.

For account-based chains (Ethereum, Lux C-Chain), every account that has sent at least one transaction has its public key permanently recorded on-chain.

4.4 Attack 4: Cross-Chain Bridge Key Extraction

Cross-chain bridges typically use MPC (multi-party computation) with ECDSA threshold signatures. The bridge's public key is on-chain. If the threshold scheme uses a (t, n) configuration and the adversary can recover any t individual signer public keys, the adversary can recover each signer's private key independently and forge bridge signatures.

This is particularly dangerous because bridge contracts typically hold large asset pools: Wormhole ($2.4B TVL), LayerZero ($1.8B TVL), and similar systems are single-point-of-failure targets.

4.5 Attack 5: Oracle Attestation Forgery

Price oracles sign attestations with ECDSA or EdDSA keys. A quantum adversary who recovers oracle signing keys can forge arbitrary price feeds, enabling:

- Manipulation of DeFi lending protocol liquidations
- Artificial arbitrage through forged price discrepancies
- Triggering of options/derivatives at manipulated strike prices

5 Timeline Estimates for Cryptographically Relevant Quantum Computers

5.1 Hardware Roadmaps

We survey published roadmaps from leading quantum hardware vendors:

Table 3: Quantum hardware roadmaps (published projections as of 2025).

Vendor	Qubits (2025)	Projected CRQC	Source
IBM	1,121 (Condor)	2029–2033	IBM Quantum Roadmap
Google	105 (Willow)	2029–2035	Google AI Blog
PsiQuantum	—	2027–2030	Photonic approach
IonQ	36 (Forte)	2030–2035	Trapped ion
QuEra	256 (neutral atom)	2028–2033	Neutral atom

5.2 Logical vs. Physical Qubits

Current quantum computers use physical qubits with error rates of $\sim 10^{-3}$. Breaking secp256k1 requires $\sim$2,330 *logical* qubits with error rates $< 10^{-10}$. Using surface codes with code distance 27, each logical qubit requires $\sim$1,457 physical qubits [4]. Therefore:

$$2{,}330 \times 1{,}457 \approx 3{,}394{,}810 \text{ physical qubits}$$

This is approximately 3,000$\times$ more physical qubits than the largest current systems. However, qubit counts have been growing exponentially, and error correction improvements may reduce the overhead factor significantly.

5.3 Probability Assessment

Based on published roadmaps, error correction progress, and the distinction between physical and logical qubits, we estimate:

Table 4: Cumulative probability of CRQC existence.

Year	**P(CRQC exists)**
2028	$< 1\%$
2030	5–10%
2033	15–30%
2035	30–50%
2040	$> 70\%$

These estimates are consistent with Mosca's inequality [5]: if the time to deploy quantum-safe cryptography (d) plus the shelf life of current secrets (l) exceeds the time until CRQCs exist (q), then action is needed now. For blockchain, $l = \infty$ (on-chain data never expires), so the condition $d < q$ must hold—meaning migration must *complete* before CRQCs exist. Given that protocol-level cryptographic migrations take 3–5 years to fully deploy, the window for action is already closing.

6 Economic Analysis

6.1 Cost of Not Migrating

We estimate the total value at risk from quantum attacks on blockchain infrastructure:

Table 5: Estimated value at risk from quantum attacks (as of Q1 2026).

Category	**Value at Risk (USD)**
Bitcoin (P2PK + reused P2PKH outputs)	\$180B
Ethereum (all accounts that sent ≥ 1 tx)	\$240B
Cross-chain bridge TVL	\$12B
DeFi protocol TVL (EVM chains)	\$85B
Stablecoin supply (ECDSA-secured)	\$160B
Total	$>$ \$500**B**

The expected loss under a 30% probability of CRQC by 2035 is:

$$E[\text{loss}] = 0.30 \times \$500\text{B} = \$150\text{B}$$

6.2 Cost of Hybrid Deployment

Table 6: Estimated cost of hybrid post-quantum deployment.

Cost Category	**Estimate**
Engineering (protocol-level changes)	\$5M–\$20M
Throughput overhead (15% reduction)	Negligible
Storage overhead (248 B/epoch)	<100 MB/year
Validator hardware upgrades	\$0 (software update)
Audit and formal verification	\$2M–\$5M
Total one-time cost	\$7M–\$25M

The cost-benefit ratio is overwhelming: \$25M investment to protect against \$150B expected loss, a ratio of 1 : 6,000.

6.3 Game-Theoretic Argument

Even if the probability of CRQC is low (say 5% by 2030), the rational strategy is still immediate deployment because:

1. The cost of deployment is fixed and bounded.
2. The cost of a successful quantum attack is unbounded (total chain compromise).
3. Hybrid deployment has no downside: it cannot make security *worse* because the classical component remains as a fallback.
4. The "wait for certainty" strategy fails because migration takes years, and by the time CRQC is certain, migration may not complete in time.

This is a classical insurance argument: the premium (deployment cost) is negligible relative to the insured value, and the policy has no exclusions (hybrid is strictly additive).

7 Recommendations

Based on our threat model, timeline analysis, and economic assessment, we make the following concrete recommendations for blockchain protocol designers:

7.1 Immediate Actions (2026)

1. **Deploy hybrid signatures**: Implement BLS + ML-DSA-65 dual signatures for validator consensus. ML-DSA signatures can be collected but not enforced initially to allow testing without consensus risk.
2. **Deploy hybrid key exchange**: Implement X25519 + ML-KEM-768 for all node-to-node encrypted communication.
3. **Register PQ public keys**: Require all validators to register ML-DSA public keys alongside BLS public keys. This enables enforcement without a second key generation ceremony.
4. **Audit hash function usage**: Verify that all hash-based constructions (Merkle trees, address derivation) use ≥256-bit output hashes.

7.2 Medium-Term Actions (2027–2028)

1. **Enforce hybrid verification**: Require both BLS and ML-DSA signatures for epoch finality. Blocks that lack valid ML-DSA epoch proofs are not considered finalized.

2. **Migrate bridge MPC**: Transition bridge threshold signature schemes from ECDSA-only to hybrid ECDSA + Pulsar (lattice-based threshold).

3. **User-facing PQ signatures**: Enable EVM account abstraction with ML-DSA signature verification for user transactions.

7.3 Long-Term Actions (2029+)

1. **Monitor quantum progress**: Track CRQC development against the timeline estimates in Section 5.

2. **Prepare for classical deprecation**: If quantum computers reach sufficient scale, deprecate BLS and ECDSA, retaining only post-quantum primitives.

3. **Address hash migration**: If Grover-based collision attacks become practical, migrate Merkle tree hashes to 512-bit outputs.

8 Conclusion

Blockchain infrastructure faces a unique quantum threat due to the permanence of on-chain data, the static nature of validator keys, and the persistent economic value protected by cryptographic signatures. The harvest-now-decrypt-later attack model means the threat is active *today*, not in some hypothetical future when quantum computers mature.

The economic analysis is unambiguous: the cost of hybrid deployment is negligible relative to the value at risk. The game theory is clear: hybrid deployment is strictly dominant—it cannot reduce security and may be the difference between survival and catastrophic loss. The only losing strategy is to wait for certainty before acting.

References

[1] P. W. Shor, "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer," *SIAM Journal on Computing*, vol. 26, no. 5, pp. 1484–1509, 1997.

[2] L. K. Grover, "A Fast Quantum Mechanical Algorithm for Database Search," *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, pp. 212–219, 1996.

[3] M. Roetteler, M. Naehrig, K. M. Svore, and K. Lauter, "Quantum Resource Estimates for Computing Elliptic Curve Discrete Logarithms," *Advances in Cryptology – ASIACRYPT 2017*, pp. 241–270, Springer, 2017.

[4] C. Gidney and M. Ekerå, "How to Factor 2048 Bit RSA Integers in 8 Hours Using 20 Million Noisy Qubits," *Quantum*, vol. 5, p. 433, 2021.

[5] M. Mosca, "Cybersecurity in an Era with Quantum Computers: Will We Be Ready?" *IEEE Security & Privacy*, vol. 16, no. 5, pp. 38–41, 2018.

[6] NIST, "Module-Lattice-Based Key-Encapsulation Mechanism Standard (FIPS 203)," National Institute of Standards and Technology, August 2024.

[7] NIST, "Module-Lattice-Based Digital Signature Standard (FIPS 204)," National Institute of Standards and Technology, August 2024.

[8] NIST, "Stateless Hash-Based Digital Signature Standard (FIPS 205)," National Institute of Standards and Technology, August 2024.

[9] D. Boneh, B. Lynn, and H. Shacham, "Short Signatures from the Weil Pairing," *Journal of Cryptology*, vol. 17, no. 4, pp. 297–319, 2004.

[10] S. E. Webber, S. Patil, D. Linke, et al., "The Impact of Hardware Specifications on Reaching Quantum Advantage in the Fault Tolerant Regime," *AVS Quantum Science*, vol. 4, no. 1, 2022.

[11] E. T. Campbell, B. M. Terhal, and C. Vuillot, "Roads Towards Fault-Tolerant Universal Quantum Computation," *Nature*, vol. 549, pp. 172–179, 2017.

[12] J. Preskill, "Quantum Computing in the NISQ Era and Beyond," *Quantum*, vol. 2, p. 79, 2018.