



LP-309: Quantum Network Comparison

Per-Chain Quantum
Vulnerability Analysis and the

Case for Post-Quantum Infrastructure

Zach Kelling

Lux Industries

Original: January 2025 | Revised: March 2026

Abstract

We analyze the quantum vulnerability of four major blockchain networks—Bitcoin, Ethereum, Solana, and Cardano—examining their key generation, address derivation, and transaction signing processes to determine the window of public key exposure available to a quantum adversary. We then describe how Lux Network’s architecture mitigates these vulnerabilities through post-quantum consensus signatures, threshold key management with continuous rotation, and an X-Chain UTXO model that minimizes key exposure.

This is not a hypothetical exercise. NIST estimates that cryptanalytically relevant quantum computers may emerge within 10–15 years [3]. Assets on-chain today will still be on-chain then. The time to migrate is now.

Contents

1	Introduction	3
2	Per-Chain Analysis	3
2.1	Bitcoin	3
2.1.1	Key Generation and Address Derivation	3
2.1.2	Public Key Exposure	3
2.1.3	Quantum Vulnerability Window	3
2.2	Ethereum	3
2.2.1	Key Generation and Address Derivation	3
2.2.2	Public Key Exposure	3
2.2.3	Quantum Vulnerability Window	4
2.3	Solana	4
2.3.1	Key Generation and Address Derivation	4
2.3.2	Public Key Exposure	4
2.3.3	Quantum Vulnerability Window	4
2.4	Cardano	4
2.4.1	Key Generation and Address Derivation	4
2.4.2	Public Key Exposure	4
2.4.3	Quantum Vulnerability Window	4
3	Comparative Summary	5
3.1	Key Observations	5
4	Lux Network’s Post-Quantum Architecture	5
4.1	Consensus Layer	5
4.2	X-Chain: UTXO with Minimal Exposure	5
4.3	EVM Precompiles	6
4.4	Bridge: MPC with Key Rotation	6
4.5	Hybrid Encryption: X-Wing	6
5	Applications Beyond Finance	6
6	Conclusion	7

1 Introduction

Every major blockchain network relies on elliptic curve cryptography (ECDSA over secp256k1, or EdDSA over Ed25519) for transaction signing. Shor’s algorithm [1] can solve the elliptic curve discrete logarithm problem in polynomial time on a fault-tolerant quantum computer, enabling an adversary to derive private keys from exposed public keys.

The critical question is not *whether* this is possible, but *how long* each network exposes public keys to attack. This exposure window determines the practical risk: a network where public keys are visible for seconds has a fundamentally different threat profile than one where they are visible indefinitely.

2 Per-Chain Analysis

2.1 Bitcoin

2.1.1 Key Generation and Address Derivation

Bitcoin generates an ECDSA key pair (sk, pk) on secp256k1. The address is derived through two hash layers:

1. $h_1 = \text{SHA-256}(pk)$
2. $h_{160} = \text{RIPEMD-160}(h_1)$
3. $\text{address} = \text{Base58Check}(\text{version} || h_{160} || \text{checksum})$

2.1.2 Public Key Exposure

The public key pk is *not* revealed when receiving Bitcoin. The address is a hash of pk , and hash preimage resistance protects pk even if the address is known. The public key is revealed only when *spending*—the transaction includes pk for signature verification.

2.1.3 Quantum Vulnerability Window

The exposure window is the time between transaction broadcast and block confirmation (~ 10 minutes for one confirmation, ~ 60 minutes for six). An adversary must derive sk from pk within this window to redirect funds.

Remark 2.1. *Bitcoin’s recommended practice of single-use addresses provides additional protection: once an address’s public key is revealed by spending, no further funds should be sent to it. Unspent outputs at never-used addresses have zero public key exposure.*

2.2 Ethereum

2.2.1 Key Generation and Address Derivation

Ethereum generates an ECDSA key pair on secp256k1. The address is the last 20 bytes of the Keccak-256 hash of the public key:

$$\text{address} = \text{Keccak-256}(pk)[12:32]$$

2.2.2 Public Key Exposure

The public key is **effectively always exposed**. It can be recovered from any ECDSA signature via `ecrecover`:

$$pk = \text{ecrecover}(\text{msgHash}, v, r, s)$$

Once an account has signed any transaction, the public key is permanently derivable from the blockchain history.

2.2.3 Quantum Vulnerability Window

The exposure window is **permanent** for any account that has ever sent a transaction. An adversary does not need to race against block confirmation—they can extract pk from historical transactions at leisure and compute sk offline.

Remark 2.2. *Ethereum’s account model exacerbates the risk. Unlike Bitcoin’s UTXO model where each output can use a fresh key, Ethereum accounts are long-lived and typically reuse the same key for all transactions.*

2.3 Solana

2.3.1 Key Generation and Address Derivation

Solana uses Ed25519 key pairs. Unlike Bitcoin (SHA-256 + RIPEMD-160) or Ethereum (Keccak-256), Solana applies *no hash layer at all*: the address is the raw public key, base58-encoded.

$$\text{address} = \text{base58}(pk)$$

2.3.2 Public Key Exposure

The public key is *always* visible to anyone who knows the address—there is no hash-based protection, even before first spend. This is strictly weaker than Bitcoin’s UTXO model (pk hidden until spend) and equivalent in outcome to Ethereum’s permanent exposure.

2.3.3 Quantum Vulnerability Window

Permanent from the moment an address is published. A quantum adversary with Shor’s algorithm can recover the secret key from the address directly, without needing a prior transaction. Solana has the *worst* public-key exposure profile of the four chains analysed here.

2.4 Cardano

2.4.1 Key Generation and Address Derivation

Cardano uses separate spending and staking key pairs. Addresses are derived via Blake2b-224:

$$\begin{aligned}\text{PaymentKeyHash} &= \text{Blake2b-224}(\text{PublicSpendingKey}) \\ \text{StakeKeyHash} &= \text{Blake2b-224}(\text{PublicStakingKey}) \\ \text{BaseAddress} &= \text{Bech32}(\text{NetworkID} \parallel \text{PaymentKeyHash} \parallel \text{StakeKeyHash})\end{aligned}$$

2.4.2 Public Key Exposure

The spending public key is revealed when signing a transaction. Cardano’s extended UTXO model allows address reuse, but best practice is to derive fresh addresses for each transaction.

2.4.3 Quantum Vulnerability Window

Similar to Bitcoin: the public key is exposed at signing time. However, Cardano’s faster block times (~ 20 seconds) reduce the exposure window compared to Bitcoin’s 10 minutes.

3 Comparative Summary

Property	BTC	ETH	SOL	ADA
Signature scheme	ECDSA	ECDSA	Ed25519	Ed25519
Address hash	SHA-256 + RIPEMD-160	Keccak-256	none (base58 pk)	Blake2b-224
TX model	UTXO	Account	Account	Extended UTXO
PK exposure at receive	No	No	Yes	No
PK exposure at spend	Yes	Yes	Yes	Yes
PK recoverable from history	At spend	Always	Always	At spend
Exposure window	Minutes	Permanent	Permanent (from receive)	Seconds
Recommended key reuse	No	Yes (forced)	Yes (forced)	No
PQ resistance	Partial	Low	Worst	Partial

3.1 Key Observations

1. **UTXO models are inherently more resistant** than account models because they naturally support key-per-output patterns. The public key can be kept hidden until the moment of spending.
2. **Account models force permanent exposure:** Once a key is used, it is recoverable forever. Migrating to a new key requires moving all assets, which is itself a transaction that exposes the old key.
3. **Hash function security is not the bottleneck:** SHA-256, Keccak-256, and Blake2b-224 are all believed to be quantum-resistant (Grover provides only a quadratic speedup [2], so 256-bit hashes provide ~ 128 bits of quantum security). The vulnerability is in the *signature scheme*, not the address derivation.
4. **The real target is frozen funds:** Addresses with known public keys and large balances (exchange cold wallets, long-term holders, lost key situations) are the highest-value quantum targets. These funds sit on-chain with their public keys exposed indefinitely.

4 Lux Network’s Post-Quantum Architecture

4.1 Consensus Layer

Quasar consensus (LP-020) uses a hybrid signature scheme:

- **BLS-381:** Aggregate signatures for validator voting. Efficient (~ 48 bytes per signature), enabling compact block headers. Vulnerable to quantum attack via discrete log on pairing-friendly curves.
- **ML-DSA-65** (FIPS 204): NIST-standardized lattice-based signatures. 3.3 KB signatures. Quantum-resistant under LWE hardness assumption.
- **Pulsar:** Lattice-based ring signatures providing signer anonymity within a validator set. Post-quantum by construction.

The hybrid design provides defense in depth: if BLS is broken by a quantum adversary, ML-DSA and Pulsar signatures on the same blocks prevent consensus forgery.

4.2 X-Chain: UTXO with Minimal Exposure

Lux’s X-Chain inherits the UTXO model. Each transaction output is associated with a fresh key. Public keys are revealed only at spending time, and the network’s sub-second finality (via Quasar) minimizes the exposure window to under one second.

Remark 4.1. *Compared to Bitcoin’s 10-minute exposure window, Lux’s sub-second finality reduces the quantum attack window by approximately 600×. This is not a substitute for post-quantum signatures—it is an additional defense layer.*

4.3 EVM Precompiles

The Lux EVM provides post-quantum cryptographic operations as precompiled contracts in the reserved 0x0200 range (see `lux-evm-precompile-engineering.tex` for the canonical specification, addresses, and gas costs). Available primitives include ML-DSA-65 / SLH-DSA-128s / ML-KEM-768 verify and decapsulate, FROST and CGGMP21 threshold verification, Pulsar ring verification, and TFHE evaluation gates.

Smart contracts can verify post-quantum signatures natively, without off-chain computation or trusted oracles.

4.4 Bridge: MPC with Key Rotation

The Lux Bridge uses 2-of-3 MPC threshold signing for cross-chain transactions. Key rotation occurs at every withdrawal event:

1. Validators perform distributed key generation (DKG) for a new signing key.
2. The withdrawal transaction spends from the old multisig address and sends remaining funds to a new address derived from the new key.
3. The old key’s public component is exposed for the duration of the withdrawal transaction only.

This ensures no ECDSA public key is exposed for more than the time required to confirm the withdrawal transaction. Combined with a policy of batching withdrawals (hourly or daily), the total key exposure across a year is measured in hours, not the permanent exposure of Ethereum accounts.

4.5 Hybrid Encryption: X-Wing

For key encapsulation and encrypted communication, Lux uses X-Wing (X25519 + ML-KEM-768 hybrid) implemented in `luxfi/age`. This provides post-quantum confidentiality: an adversary must break *both* the elliptic curve and lattice components to decrypt.

5 Applications Beyond Finance

Post-quantum security is critical across multiple sectors where data encrypted today must remain confidential for decades:

- **Government:** Classified document authentication, secure inter-agency communication, e-voting integrity.
- **Defense:** Secure command channels, supply chain verification, satellite communication authentication.
- **Healthcare:** Long-term patient record confidentiality, genomic data protection, cross-institution research without data exposure (via FHE).
- **Finance:** Transaction signing, key management for custodial assets, regulatory reporting with privacy guarantees.

The “harvest now, decrypt later” attack is not theoretical—it is the baseline assumption for any adversary with long-term strategic interests. Data encrypted under RSA or ECDSA today can be stored and decrypted when quantum computers become available.

6 Conclusion

The analysis demonstrates a clear hierarchy of quantum vulnerability among existing blockchain networks:

1. **Lowest risk:** UTXO-based networks with single-use addresses (Bitcoin, Lux X-Chain). Public key exposure is transient and can be minimized by key hygiene.
2. **Highest risk:** Account-based networks (Ethereum, Solana). Public keys are permanently recoverable from transaction history.

Neither category is *safe*—both rely on ECDSA or EdDSA, which fall to Shor’s algorithm. The defense requires replacing the signature scheme itself, not merely minimizing exposure windows.

Lux Network’s approach is the only production architecture that addresses all three layers: post-quantum *consensus* signatures (ML-DSA + Pulsar), post-quantum *key encapsulation* (X-Wing), and post-quantum *smart contract verification* (EVM precompiles). The UTXO model and sub-second finality provide additional defense in depth, but they are not substitutes for correct cryptography.

References

- [1] P. W. Shor, “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer,” *SIAM Review*, vol. 41, no. 2, pp. 303–332, 1999.
- [2] L. K. Grover, “A fast quantum mechanical algorithm for database search,” *STOC*, pp. 212–219, 1996.
- [3] NIST, “Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process,” NISTIR 8413, 2022.
- [4] NIST, “Post-Quantum Cryptography Standardization,” <https://csrc.nist.gov/projects/post-quantum-cryptography>, 2024.
- [5] D. J. Bernstein and T. Lange, “Post-quantum cryptography—dealing with the fallout of physics success,” *IACR Cryptology ePrint Archive*, 2017/314.
- [6] O. Regev, “On lattices, learning with errors, random linear codes, and cryptography,” *STOC*, pp. 84–93, 2005.
- [7] D. Micciancio and O. Regev, “Lattice-based cryptography,” in *Post-Quantum Cryptography*, pp. 147–191, Springer, 2009.
- [8] V. Lyubashevsky, C. Peikert, and O. Regev, “On ideal lattices and learning with errors over rings,” *Journal of the ACM*, vol. 60, no. 6, pp. 1–43, 2013.
- [9] Z. Kelling, “LP-020: Quasar—Hybrid BLS + Lattice Consensus for Post-Quantum Finality,” Lux Industries, 2026.
- [10] Z. Kelling, “LP-305: Quantum Secure Assets—A Post-Quantum Methodology for Protecting Digital Assets on the Lux Network,” Lux Industries, 2022–2026.