

Cryptocurrency Transaction Map: Quantum Exposures (DRAFT)

Critical stages where Post-Quantum Cryptography (PQC) and Quantum Key Distribution (QKD) are needed

Contents

Client-Side Security Layer • Network Communication Layer • Exchange Platform Security Layer • Blockchain Integration Layer •
Quantum Vulnerability & Implementation Priority • Service Providers where QKD is Needed • Technical Glossary

1 Client-Side Security Layer

MEDIUM RISK

User Account Authentication

Password hashes (e.g., bcrypt) and HMAC-based TOTP both affected by Grover's algorithm. RSA-based backup codes or authentication tokens vulnerable to Shor's algorithm. Primary entry point for quantum-enhanced attacks.

PQC: SHA-512 for password hashing, ML-DSA for 2FA authentication protocols

HIGH RISK

Device Authentication & Key Storage

Private keys stored on user devices, wallet applications. Vulnerable to quantum attacks on ECC and RSA key generation/storage.

PQC: ML-KEM key encapsulation, quantum-resistant signatures

HIGH RISK

Key Recovery/Social Recovery

ECDSA multisig or RSA-encrypted backups vulnerable to Shor's algorithm. Critical for user experience and long-term security.

PQC: ML-DSA for multisig recovery, ML-KEM for encrypted backups

MEDIUM RISK

Browser/App Security

Client-side encryption, session management, local data protection. Modern browsers implement TLS but need PQC updates.

PQC: Hybrid TLS 1.3 with ML-KEM

2 Network Communication Layer

HIGH RISK

TLS/SSL Handshake

ECDHE key exchange for session keys. Current implementations use quantum-vulnerable elliptic curve cryptography.

PQC: Hybrid key exchange (Classical + ML-KEM)

QKD: Quantum key distribution for ultra-high security

HIGH RISK

Certificate Validation

X.509 certificates with RSA/ECDSA signatures for server authentication. PKI infrastructure vulnerable to quantum attacks.

PQC: ML-DSA signatures, SLH-DSA for long-term security

LOW RISK

API Authentication

OAuth/HMAC uses symmetric cryptography, less vulnerable to quantum attacks. Grover's algorithm reduces HMAC key search strength (e.g., 256-bit to 128-bit effective). Asymmetric token signing (e.g., RSA) is vulnerable to Shor's algorithm.

PQC: 256-bit HMAC keys (sufficient vs Grover's), quantum-resistant token signing

3 Exchange Platform Security Layer

HIGH RISK

Third-Party Custodial Services

RSA/ECDSA-based authentication vulnerable to Shor's algorithm. Critical for institutional exchanges with inter-party authentication requirements.

PQC: ML-DSA/ML-KEM for custodial authentication and key exchange

QKD: Applicable for secure key handoffs in controlled environments

HIGH RISK

Hot Wallet Management

Real-time transaction signing and multi-signature schemes using ECDSA signatures vulnerable to quantum attacks. Vulnerability during API communications with exchanges, transaction signing, and broadcasting.

PQC: ML-DSA multi-sig, quantum-resistant threshold signatures

QKD: Theoretically possible but limited by infrastructure (dedicated quantum channels)

HIGH RISK

Cold Storage Security

Offline private key storage using hardware security modules and air-gapped systems. Keys encrypted with RSA/ECC vulnerable to future quantum decryption (harvest now, decrypt later). Critical exposure during key generation ceremonies, multi-location backup distribution, and recovery procedures.

PQC: ML-KEM key encapsulation

QKD: Quantum key distribution for key management

LOW RISK

Database Encryption

AES is moderately affected by Grover's algorithm (128-bit AES reduced to 64-bit effective strength). Asymmetric key derivation (e.g., via RSA) is vulnerable to Shor's algorithm.

PQC: 256-bit AES (quantum-resistant), quantum-resistant key derivation protocols

MEDIUM RISK

Inter-Service Communication

Microservices authentication, service mesh security, internal API protection.

PQC: Post-quantum mutual TLS, service identity certificates

4 Blockchain Integration Layer

HIGH RISK

Transaction Signing

ECDSA signatures for Bitcoin/Ethereum transactions. Core blockchain cryptography vulnerable to quantum attacks.

PQC: Requires blockchain protocol upgrades to quantum-resistant signatures

LOW RISK

Proof of Work Consensus

SHA-256-based PoW (e.g., Bitcoin) is moderately affected by Grover's algorithm, reducing hash search strength (256-bit to 128-bit effective). Quantum miners could gain advantage, potentially enabling 51% attacks, though this could be mitigated by network difficulty adjustments or transitions to quantum-resistant consensus like Proof of Stake.

PQC: SHA-512/SHA-3 for increased hash strength, consider Proof of Stake transition

MEDIUM RISK

DEX Peer Authentication

ECDSA-based peer authentication vulnerable to Shor's algorithm. Growing importance as DEXs expand in the crypto ecosystem.

PQC: ML-DSA for peer signatures in DEX protocols

MEDIUM RISK

Smart Contract Security

Contract deployment, execution, and interaction signatures. Ethereum uses ECDSA for transaction authorization.

PQC: Post-quantum smart contract platforms, upgraded EVM




LOW RISK



Block & Data Verification

Hash functions (SHA-256, Keccak) for block hashing, Merkle trees.
Hash functions moderately affected by Grover's algorithm (square root speedup), but remain secure with doubled output sizes.

PQC: Increase to SHA-512 or SHA-3 (double bit strength vs Grover's)

Quantum Vulnerability & Implementation Priority

 **High Risk:** Immediate PQC/QKD needed  **Medium Risk:** PQC recommended  **Low Risk:** Monitor and upgrade when feasible

Solution Types:  **Post-Quantum Cryptography:** Mathematical algorithms  **Quantum Key Distribution:** Physics-based quantum security

Key PQC Standards (NIST-approved):

- **ML-KEM:** Key encapsulation mechanism (successor to CRYSTALS-Kyber)
- **ML-DSA:** Digital signatures (successor to CRYSTALS-Dilithium)
- **SPHINCS+:** Stateless hash-based signatures for long-term security
- **SLH-DSA:** Alternative signature algorithm based on hash functions
- **HQC:** Hamming Quasi-Cyclic code-based key encapsulation (NIST backup standard, March 2025)

Main Players/Service Providers Where QKD is Needed

TLS/SSL Network Communications

Service Providers: CloudFlare, AWS ALB/NLB, Google Cloud Load Balancing, Akamai
Use Case: Ultra-high security quantum key distribution for critical API endpoints
Implementation: QKD networks between data centers and high-value client connections

Regulatory/Government Interfaces

Players: Chainalysis, Elliptic, TRM Labs (compliance), central bank digital currencies (CBDCs)
Use Case: Quantum-secured reporting and compliance data transmission
Implementation: QKD for government oversight and regulatory reporting

Third-Party Custodial Services

Players: Coinbase Custody, BitGo, Fireblocks, Anchorage Digital, Bakkt
Use Case: Secure key handoffs between institutional custody providers
Implementation: QKD for inter-custodian transfers and institutional client onboarding

Cold Storage Security

Players: Ledger Enterprise, Casa, Unchained Capital, Copper.co
Use Case: Quantum-secured key management for offline storage systems
Implementation: QKD for key ceremony processes and secure backup distribution

High-Value Transaction Networks

Players: Circle (USDC), Tether (USDT), institutional DEXs like dYdX, cross-chain bridges
Use Case: Ultra-secure channels for large institutional transactions
Implementation: Dedicated quantum channels for transactions >\$100M

Technical Glossary

Cryptographic Terms

AES: Advanced Encryption Standard - symmetric encryption algorithm

ECC: Elliptic Curve Cryptography - public key cryptography using elliptic curves

ECDHE: Elliptic Curve Diffie-Hellman Ephemeral - key exchange protocol

ECDSA: Elliptic Curve Digital Signature Algorithm - digital signature scheme

HMAC: Hash-based Message Authentication Code - message authentication

HSM: Hardware Security Module - dedicated crypto processor

PKI: Public Key Infrastructure - framework for managing digital certificates

RSA: Rivest-Shamir-Adleman - public key cryptosystem

SHA: Secure Hash Algorithm - cryptographic hash function family

Network & Protocol Terms

API: Application Programming Interface - software interface

EVM: Ethereum Virtual Machine - runtime environment for smart contracts

OAuth: Open Authorization - authorization framework

REST: Representational State Transfer - architectural style for web services

SSL/TLS: Secure Sockets Layer / Transport Layer Security - protocols for secure communication

WebSocket: Communication protocol for full-duplex communication

X.509: Standard for public key certificates

Post-Quantum Cryptography

Grover's Algorithm: Quantum algorithm that provides quadratic speedup for searching unsorted databases, affecting symmetric cryptography

HQC: Hamming Quasi-Cyclic - code-based key encapsulation mechanism, NIST backup standard

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

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

NIST: National Institute of Standards and Technology

PQC: Post-Quantum Cryptography - algorithms secure against quantum attacks

QKD: Quantum Key Distribution - quantum physics-based key sharing

Shor's Algorithm: Quantum algorithm that efficiently factors large integers and solves discrete logarithms, breaking RSA and ECC

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

SPHINCS+: Hash-based signature scheme

Blockchain & Exchange Terms

2FA: Two-Factor Authentication - security method requiring two verification factors

51% Attack: Attack where a single entity controls majority of network's mining power

bcrypt: Password hashing function based on the Blowfish cipher

CEX: Centralized Exchange - traditional exchange with central authority managing trades and funds

Cold Storage: Offline cryptocurrency storage for security

DEX: Decentralized Exchange - peer-to-peer trading platform

Hot Wallet: Online cryptocurrency wallet for active trading

Keccak: Cryptographic hash function used in Ethereum

Merkle Tree: Binary tree structure for efficient data verification

Multi-sig: Multi-signature - requiring multiple signatures for transactions

PoS: Proof of Stake - consensus mechanism based on stake ownership rather than computational work

PoW: Proof of Work - consensus mechanism requiring computational effort to validate blocks

Smart Contract: Self-executing contract

with terms in code

TOTP: Time-based One-Time Password -
algorithm for generating temporary codes

Dmitry Green AppliedTQC.com