

WHITE PAPER · OCTOBER 2026

Post-Quantum Migration: A Practical Playbook for Engineering Teams

How harvest-now-decrypt-later turns a future threat into a present deadline, and a four-step method for moving production systems to NIST's post-quantum standards.



Kyeson Blake Utley

Founder, GhostKey Development

ORCID 0009-0001-4531-2062 · kyesonutley.com



Abstract

Public-key cryptography based on RSA and elliptic curves will not survive a cryptographically relevant quantum computer. Because adversaries can record encrypted traffic today and decrypt it later, data with a long secrecy lifetime is already at risk. This paper summarizes the threat model, the post-quantum standards NIST published in 2024, and the official deprecation timelines, then sets out a four-step migration method (inventory, prioritization by secrecy lifetime, hybrid deployment and crypto-agility) along with the engineering constraints teams meet in practice. It closes with considerations specific to cryptocurrency systems and a migration checklist.

1. Introduction

Most organizations treat post-quantum cryptography as a future project. The usual reasoning is that no quantum computer capable of breaking today's public-key cryptography exists yet, so there is time. That reasoning misreads the threat. The relevant question is not when such a machine will exist, but how long today's encrypted data needs to stay secret, and how long the migration will take.

This paper is written for engineering and security leads who need to turn that question into a plan. It draws on GhostKey Development's work building and auditing cryptographic and cryptocurrency systems.

2. Threat model: harvest now, decrypt later

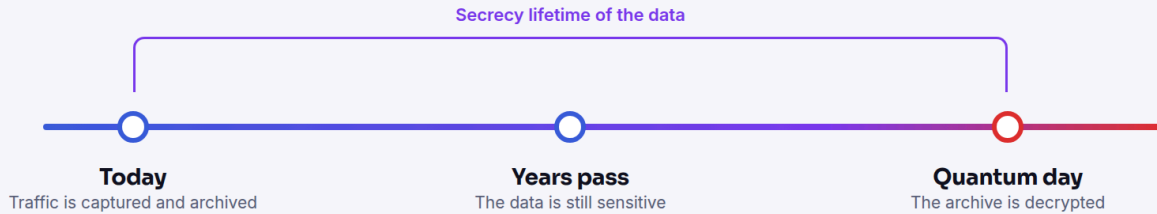
Shor's algorithm, run on a sufficiently large fault-tolerant quantum computer, solves the integer factorization and discrete logarithm problems efficiently. That breaks RSA, finite-field Diffie-Hellman and elliptic-curve schemes (ECDH, ECDSA, EdDSA). Symmetric ciphers and hash functions are affected far less: Grover's algorithm gives at most a quadratic speed-up, which larger key and output sizes (for example AES-256 and SHA-384) absorb.

The practical consequence is that key exchange is the urgent problem. An adversary who records a TLS or VPN session today can store it indefinitely. Once a quantum computer can recover the session's key exchange, the recorded traffic can be decrypted, regardless of how strong the symmetric encryption was.

POST-QUANTUM RISK

Harvest now, decrypt later

Encrypted traffic recorded today can be stored until a quantum computer can break it.



If **how long data must stay secret** + **how long migration takes** > **time until quantum**, that data is already exposed.

Diagram by Kyeson Blake Utley · kyesonutley.com

Figure 1. Traffic captured today stays sensitive for years and becomes readable once a quantum computer arrives.

Mosca's inequality

Michele Mosca framed the decision as a simple inequality. Let x be how long data must remain confidential, y how long it will take to migrate the systems protecting it, and z the time until a cryptographically relevant quantum computer exists. If $x + y > z$, the data is already exposed. Estimates of z vary widely; the inequality matters because x and y are under an organization's control and are often longer than teams assume.

3. The standards landscape

In August 2024 NIST published its first three post-quantum standards. In March 2025 it selected HQC as a backup key-encapsulation mechanism based on different mathematics.

Standard	Algorithm	Use
FIPS 203	ML-KEM (Module-Lattice Key Encapsulation)	Key establishment; replaces RSA and ECDH key exchange
FIPS 204	ML-DSA (Module-Lattice Digital Signature)	General-purpose signatures
FIPS 205	SLH-DSA (Stateless Hash-Based Signature)	Conservative signatures relying only on hash functions
HQC (selected 2025)	Code-based key encapsulation	Backup to ML-KEM

Official timelines

NIST IR 8547 (initial public draft, November 2024) proposes deprecating quantum-vulnerable public-key algorithms at the 112-bit security level after 2030 and disallowing quantum-vulnerable algorithms entirely after 2035. The NSA's Commercial National Security Algorithm Suite 2.0 (CNSA 2.0) sets transition milestones for national security systems through the early 2030s, consistent with full quantum resistance by 2035. Organizations that sell to government or regulated industries should expect these dates to flow down through procurement requirements.

4. A four-step migration method

Step 1: Inventory

Build a cryptographic inventory before changing anything. Record every place the organization performs key exchange or signing: TLS terminators and load balancers, VPNs, SSH, code and firmware signing, certificate authorities, hardware security modules, service meshes, messaging, backups and, for crypto businesses, wallet and custody software. For each, note the library and version, the algorithms in use, who owns it and what data it protects. Software bills of materials and network scans help, but expect manual work; cryptography hides in dependencies.

Step 2: Prioritize by secrecy lifetime

Rank systems by the value of x in Mosca's inequality, not by how central they feel. A rarely used archive replication link carrying records that must stay confidential for decades outranks a high-traffic API whose data is stale within days. Signing systems deserve separate treatment: forged signatures matter only after a quantum computer exists, but

long-lived trust anchors such as firmware-signing keys and root certificates take years to rotate, so their migration time y is large.

Step 3: Deploy hybrid key exchange

Combine a classical algorithm with a post-quantum one so that a session stays secure as long as either holds. In TLS 1.3 the common construction pairs X25519 with ML-KEM-768 (standardized in the IETF as X25519MLKEM768). Major browsers, content delivery networks and messaging apps have already deployed hybrid key exchange, which makes it the lowest-risk first move for internet-facing systems. Recent library releases, including OpenSSL 3.5, ship ML-KEM, ML-DSA and SLH-DSA.

Step 4: Build crypto-agility

Treat this migration as the first of several. Isolate cryptographic choices behind well-defined interfaces and configuration, avoid hard-coding algorithms or key sizes, and make sure systems can rotate keys and certificates quickly. Agility is what turns the next algorithm change from a multi-year project into a configuration update.

5. Engineering constraints

Post-quantum algorithms are fast, but their keys, ciphertexts and signatures are much larger than their classical counterparts.

Item	Classical	Post-quantum
Key-exchange public key	X25519: 32 bytes	ML-KEM-768: 1,184 bytes
Key-exchange ciphertext	X25519: 32 bytes	ML-KEM-768: 1,088 bytes
Signature	Ed25519: 64 bytes	ML-DSA-65: 3,309 bytes
Signature public key	Ed25519: 32 bytes	ML-DSA-65: 1,952 bytes

- **Handshake size.** Larger key shares can push a TLS ClientHello across packet boundaries. Some middleboxes and older network equipment mishandle this; test on real network paths.
- **Certificate chains.** Post-quantum signatures enlarge certificates and chains, which affects latency and constrained devices. Plan signature migration separately from key exchange.

- **Hardware support.** Confirm HSM, smart-card and secure-element support before committing to a design; firmware updates may be required.
- **Side channels.** New implementations bring new timing and power side channels. Prefer mature, audited libraries and verify constant-time behavior in the compiled binary, not only in source.
- **Downgrade paths.** Hybrid deployments must not silently fall back to classical-only key exchange when an attacker interferes with negotiation.

6. Considerations for cryptocurrency systems

Most blockchains authorize transactions with elliptic-curve signatures such as ECDSA or Schnorr over secp256k1, or Ed25519. These are vulnerable to Shor's algorithm. The exposure depends on whether a public key is visible: funds held at addresses whose public keys are already on-chain, for example because the address has been spent from, reused, or uses an output type that embeds the public key directly, are at greater risk than funds behind an unrevealed hash of a key.

Custody providers cannot change a chain's signature scheme on their own, but they can reduce exposure now: avoid address reuse, keep cold-storage public keys unrevealed where the protocol allows it, use threshold and multi-party signing so that no single key controls large balances, and track the post-quantum proposals for the chains they support. Encrypted off-chain data such as key backups and customer records should follow the same hybrid key-exchange guidance as any other long-lived secret.

KEY CUSTODY

One key versus a 2-of-3 threshold

Single key

One phishing page, one stolen device or one coerced person moves everything.



2-of-3 threshold

Signing needs two independent keys. Losing any one is survivable.

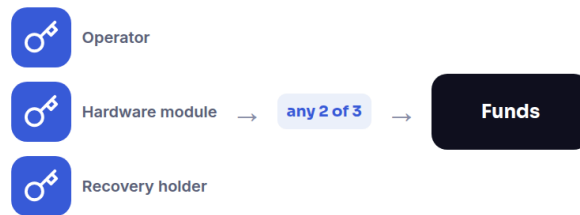


Diagram by Kyeson Blake Utley · kyesonutley.com

Figure 2. Threshold signing limits the damage any single key compromise can do, classical or quantum.

7. Migration checklist

1. Assign an owner and budget for the cryptographic migration.
2. Build and maintain a cryptographic inventory, including dependencies.
3. Estimate the secrecy lifetime of each data class and the migration time of each system.
4. Enable hybrid ML-KEM key exchange on internet-facing TLS first.
5. Plan signature and certificate migration, starting with long-lived trust anchors.
6. Confirm library, HSM and vendor post-quantum support and roadmaps.
7. Test handshakes over real networks; monitor for failures and downgrades.
8. Refactor toward crypto-agility so algorithms can be changed by configuration.
9. Audit implementations for side channels and integration errors.
10. Review progress against the 2030 and 2035 milestones annually.

8. Conclusion

The quantum threat to public-key cryptography is unusual: its damage can begin before the machine that causes it exists. Organizations that inventory their cryptography now, deploy hybrid key exchange where secrecy lifetimes are long, and build for agility will meet the 2030 and 2035 milestones as routine engineering rather than an emergency.

References

1. NIST. FIPS 203: Module-Lattice-Based Key-Encapsulation Mechanism Standard. August 2024. <https://csrc.nist.gov/pubs/fips/203/final>
2. NIST. FIPS 204: Module-Lattice-Based Digital Signature Standard. August 2024. <https://csrc.nist.gov/pubs/fips/204/final>
3. NIST. FIPS 205: Stateless Hash-Based Digital Signature Standard. August 2024. <https://csrc.nist.gov/pubs/fips/205/final>
4. NIST. IR 8547 (Initial Public Draft): Transition to Post-Quantum Cryptography Standards. November 2024. <https://csrc.nist.gov/pubs/ir/8547/ipd>
5. NIST. NIST Selects HQC as Fifth Algorithm for Post-Quantum Encryption. March 2025. <https://www.nist.gov/news-events/news/2025/03/nist-selects-hqc-fifth-algorithm-post-quantum-encryption>
6. NSA. Announcing the Commercial National Security Algorithm Suite 2.0. September 2022. https://media.defense.gov/2022/Sep/07/2003071834/-1/-1/0/CSA_CNSA_2.0_ALGORITHMS_.PDF
7. IETF. Post-quantum hybrid ECDHE-MLKEM Key Agreement for TLS 1.3 (draft-ietf-tls-ecdhe-mlkem). <https://datatracker.ietf.org/doc/draft-ietf-tls-ecdhe-mlkem/>
8. Mosca, M. Cybersecurity in an Era with Quantum Computers: Will We Be Ready? IEEE Security & Privacy 16 (5), 2018. <https://doi.org/10.1109/MSP.2018.3761723>
9. OpenSSL. OpenSSL 3.5 Final Release. April 2025. <https://openssl-library.org/post/2025-04-08-openssl-35-final-release/>
10. Google Security Blog. A new path for Kyber on the web. September 2024. <https://security.googleblog.com/2024/09/a-new-path-for-kyber-on-web.html>

Kyeson Blake Utley is the founder of GhostKey Development (ghostkey.dev), a Texas engineering firm specializing in post-quantum cryptography, cryptocurrency engineering and in-house penetration testing. More at kyesonutley.com and kyesonblakeutley.com. Contact: contact@ghostkey.dev.