BLOCKCHAIN® IN HEALTHCARE TODAY
PLATFORM APPROACHES IN HEALTHCARE

ORIGINAL RESEARCH

Quantum Threats to Bitcoin, Cryptocurrency and Blockchain

Rubayat Khan, PhD[1], Mazharul Karim, PhD[2] and Don Roosan, PharmD, PhD[3]

[1]Genomics Core Facility, Office of the Vice Chancellor for University of Nebraska Medical Center, Omaha, NE, USA; [2]Environmental Science and Engineering, University of Texas at El Paso, El Paso, TX, USA; [3]Computer Science, School of Engineering and Computational Sciences, Merrimack College, North Andover, MA, USA

DOI: https://doi.org/10.30953/bhty.v9.496

Corresponding Author: Dr. Don Roosan, Email: roosand@merrimack.edu

Keywords: bitcoin, blockchain security, post-quantum cryptography, quantum computing, ML-KEM, ML-DSA, SLH-DSA, healthcare blockchain

Abstract

Objective: The authors evaluated how quantum computing threatens the cryptographic primitives used in Bitcoin and other blockchain systems. These findings were translated into a standards-aligned post-quantum migration profile for healthcare ledgers, including consent, identity, provenance, audit, and encrypted off-chain data exchange.

Methodology: Narrative analysis, theoretical security analysis, and healthcare-oriented deployment mapping of classical public-key and hash primitives used in blockchain protocols were paired with an implementation-oriented migration profile based on finalized National Institute of Standards and Technology (NIST) post-quantum standards. We summarize the mathematical assumptions underlying RSA, Elliptic Curve Cryptography/Elliptic Curve Digital Signature Algorithm (ECC/ECDSA), and Secure Hash Algorithm (SHA-2/SHA-3-family) hash functions; analyze their susceptibility to Shor's and Grover's quantum algorithms; compare Federal Information Processing Standards (FIPS) 203 Module-Lattice-Based Key-Encapsulation Mechanism Standard (ML-KEM), FIPS 204 Module-Lattice-Based Digital Signature Standard (ML-DSA), and FIPS 205 Stateless Hash-Based Digital Signature Standard (SLH-DSA); and map their distinct roles to healthcare-ledger authorization, auditability, identity, and encrypted off-chain exchange. The term Advanced Hybrid Module-LWE & Code-Based (AHMC) is used only as shorthand for a standards-aligned hybrid post-quantum cryptography (PQC) migration profile and does not denote a proprietary product, a novel algorithm, or a new cryptographic primitive. Proposed adoption of hybrid, quantum-resistant cryptographic primitives for cryptocurrency wallets, transaction signatures, and ledger security during an interim migration period (hybrid classical + PQC, followed by PQC-only). Qualitative and implementation-oriented assessment of (1) break feasibility of RSA/ECC under Shor's algorithm, (2) effective security reduction for hash functions under Grover's algorithm, (3) security assumptions and composition requirements of a standards-aligned hybrid migration profile, (4) transaction-size and verification-cost impact, and (5) healthcare-specific implications for long-retention consent, identity, provenance, and audit records.

Results: Shor's algorithm reduces integer factorization and discrete logarithms to polynomial time, directly compromising Rivest–Shamir–Adleman (RSA) and ECC/ECDSA once fault-tolerant, large-scale quantum computers exist. Grover's algorithm yields a quadratic speedup for brute-force search, effectively halving the security margin of symmetric keys and hash functions at fixed output sizes. The AHMC-L1/L3/L5 profiles use ML-KEM-512/768/1024 for key establishment and ML-DSA-44/65/87 for transaction authentication, with SLH-DSA as a hash-based fallback. During a hybrid ECDSA+PQC migration, verification requires one classical and one PQC verification per authorization; transaction-size overhead is dominated by PQC signatures, approximately 2.4 KB for ML-DSA-44, 3.3 KB for ML-DSA-65, and 4.6 KB for ML-DSA-87 before script and encoding overhead. For healthcare ledgers, these findings support selective use of post-quantum signatures for:

- high-value state transitions,
- one-time public-key registration where possible, and
- continued off-chain storage of protected health information.

Deployment suitability remains contingent on workflow-specific latency, storage, availability, key lifecycle, and side-channel testing.

Conclusions: Quantum risk to blockchain signatures has direct implications for healthcare systems that depend on long-lived consent, identity, provenance, and audit records. A staged, standards-aligned migration profile can preserve authorization and ledger verifiability while keeping protected health information off-chain. The AHMC label refers only to this migration profile, not to a new cryptographic primitive; healthcare adoption requires open implementations, empirical benchmarking, crypto-agile key governance, and side-channel-resistant engineering.

Plain Language Summary

Quantum computers can solve some math problems far faster than today's computers. That matters because Bitcoin and many other blockchains rely on hard math problems to prove who owns an account and to prevent fraud. In particular, Bitcoin's digital signatures are based on elliptic curve cryptography (ECDSA). A sufficiently powerful quantum computer running Shor's algorithm could, in principle, work backwards from a public key to the private key and then create fake signatures. That would let an attacker spend funds they do not own.

Quantum computers could also make some brute-force searches faster via Grover's algorithm, which reduces the safety margin of common hash functions unless parameters are increased.

The results reported here explain which parts of cryptocurrency security are most exposed to quantum attacks and why. It then proposes a practical staged migration path that uses post-quantum cryptography alongside existing methods during a transition period. In this revision, AHMC is not presented as a new cryptographic primitive but rather as a migration architecture that can be instantiated with NIST-standardized ML-KEM for encrypted key establishment where needed, ML-DSA as the primary post-quantum transaction signature, and SLH-DSA or future code-based alternatives as fallback/diversity options. This distinction matters because PQC signatures are much larger than ECDSA signatures, so blockchain fees, block size, and node verification workload must be measured before deployment.

The key takeaway is that organizations using distributed ledgers for consent, audit, provenance, identity, or health data exchange should inventory quantum-vulnerable signatures now and test standardized post-quantum alternatives before urgent migration is required. This article uses Bitcoin and cryptocurrency as a high-stress technical benchmark, but the healthcare implication is distinct; the priorities include the following:

- preserving the long-term authenticity of signed records,
- keeping protected health information off-chain, and
- ensuring that migration does not disrupt clinical or administrative workflows.

Key Takeaways

- Quantum computing creates a long-term threat to the RSA-and ECC-based signatures used to authorize transactions and preserve trust in distributed ledgers.
- The relevant risks for healthcare include future verifiability of consent changes, access decisions, identity credentials, provenance events, and audit evidence.
- Long-term validation is important in healthcare because the evidentiary value of an event may extend beyond the active lifetime of the original key, certificate, software platform, or ledger implementation.

Submitted: February 19, 2026, Accepted: August 1, 2026, Published: August 31, 2026

Cryptocurrencies have become a widely adopted form of decentralized digital value transfer, with Bitcoin serving as the canonical example of a peer-to-peer electronic cash system.[1] Most cryptocurrency systems implement a blockchain-style distributed ledger secured by cryptography and consensus, enabling parties to transact without a centralized authority.[2–5] In these systems, cryptography is not ancillary; it authenticates ownership via digital signatures, links blocks through cryptographic hashes, and provides integrity guarantees that support the immutability of the ledger.[2,3] These guarantees have historically relied on the presumed intractability of specific mathematical problems under classical computation.[6–12] Quantum computing challenges this security model.[13–16] As quantum hardware scales, quantum algorithms can provide decisive advantages against specific cryptographic problem classes. Shor's algorithm can efficiently factor large integers and solve discrete logarithms, directly undermining Rivest–Shamir–Adleman (RSA) and elliptic-curve–based public-key cryptography.[17–20] Grover's algorithm provides a quadratic speedup for brute-force search, reducing security margins for symmetric primitives and cryptographic hash functions at fixed output sizes.[21–24] The combination threatens the long-term viability of cryptocurrency security assumptions if protocol upgrades are not proactively planned.[25–28]

Most major cryptocurrencies rely on three core cryptographic building blocks. First, public-key signatures, typically ECDSA or related elliptic-curve signatures, authorize transactions and prove control of private keys, with security grounded in the hardness of the elliptic curve discrete logarithm problem (ECDLP).[8,9,11,12] Second, some systems and supporting infrastructure use RSA for key management, legacy interoperability, or ancillary services, relying on the assumed hardness of integer factorization.[7] Third, cryptographic hash functions are used for proof-of-work, block chaining, Merkle trees, and address derivation.[29,30] Under Shor's algorithm, the discrete logarithm problem on elliptic curves becomes polynomial-time on a sufficiently large, fault-tolerant quantum computer.[19,20] In practice, once a public key Q is revealed, where Q = dG for private scalar d and base point G, a quantum adversary could recover d and forge signatures. For cryptocurrency systems, forged signatures translate into unauthorized spending and collapse of the transaction authorization model.[27] RSA-based mechanisms are similarly vulnerable: once the modulus n is factored into its prime components, the private key can be derived and signatures forged or ciphertexts decrypted.[31] Hash-based mechanisms are more resilient but still weakened. Grover's algorithm reduces the cost of a generic preimage search for an n-bit hash from O(2^n) to roughly O(2^(n/2)), motivating longer hashes or adjusted parameters for long-term security.[23,24]

The standards context has changed materially. National Institute of Standards and Technology (NIST)'s initial PQC standardization reports are now historical background rather than the current endpoint for deployment decisions.[32–36] In August 2024, NIST finalized the first three Federal Information Processing Standards for PQC: Federal Information Processing Standards (FIPS) 203, the Module-Lattice-Based Key-Encapsulation Mechanism Standard (ML-KEM); FIPS 204, the Module-Lattice-Based Digital Signature Standard (ML-DSA); and FIPS 205, the Stateless Hash-Based Digital Signature Standard (SLH-DSA).[37–39] Additional candidates may continue to be evaluated as backups or alternatives, but this manuscript now treats ML-KEM, ML-DSA, and SLH-DSA as the concrete baseline for comparison rather than referring generally to an ongoing standardization process.

These finalized standards change the blockchain migration question. Blockchains no longer need to discuss PQC only as

an open research competition; they must compare standardized algorithms with existing ECDSA/Schnorr deployments in terms of security level, signature size, public-key size, verification cost, key-management complexity, and governance risk.[40–42] ML-KEM is relevant to encrypted channels, custody workflows, wallet-to-wallet secure messaging, off-chain data exchange, and permissioned blockchain membership channels.[37,43–45] Bitcoin-like transaction authorization, however, requires digital signatures, so ML-DSA and SLH-DSA are the immediate NIST-standardized comparators for replacing or augmenting ECDSA.[38,39,46,47]

Accordingly, this work focuses AHMC as a conceptual migration architecture and protocol profile, not as a novel cryptographic primitive.[48] Production-oriented AHMC deployments should instantiate publicly reviewed, standardized components wherever possible. The previously described code-based component is treated as a research-level diversity option, not as a production transaction-signature recommendation, unless and until a specific code-based signature or related mechanism receives comparable cryptanalytic review, implementation maturity, and standardization status.[49–52]

This innovative work does not treat Bitcoin, public cryptocurrencies, and healthcare ledgers as operationally equivalent. Bitcoin is used as a mature adversarial benchmark because its public keys, transaction signatures, fee constraints, and consensus rules expose the practical costs of cryptographic migration. The healthcare translation is narrower and workflow-specific: permissioned ledgers may support consent status, access logs, identity credentials, clinical-trial provenance, pharmaceutical traceability, and integrity proofs for off-chain electronic health record data. The shared issue is long-term dependence on public-key signatures; the deployment priorities differ because healthcare places greater weight on confidentiality, identity governance, availability, revocation, audit retention, and continuity of care than on permissionless settlement and mining economics.[53,54]

Accordingly, this paper uses cryptocurrency systems to characterize quantum failure modes and transaction-level overhead, then maps those findings to a healthcare-specific migration sequence. It evaluates a standards-aligned hybrid PQC migration profile based on FIPS 203 ML-KEM, FIPS 204 ML-DSA, and FIPS 205 SLH-DSA. For continuity with earlier versions, the label AHMC is retained only as shorthand for this profile; it is not presented as a proprietary algorithm, a novel cryptographic primitive, or an independent security assumption. The analysis provides concrete parameter profiles, formalizes the required authorization properties, quantifies transaction-size and verification implications, and identifies the healthcare workflows and implementation tests needed before deployment.[37–39,53,54] The quantum threat landscape for traditional blockchain cryptography is summarized in Figure 1.

Methods

Study Design and Scope

This manuscript is a narrative, theoretical security analysis, and implementation-oriented cost model of the cryptographic primitives that secure cryptocurrency transactions and blockchains. We analyze (1) RSA-based public-key cryptography, (2) elliptic curve cryptography and ECDSA signatures, and (3) cryptographic hash functions used for block linking and proof-of-work. We then evaluate the impact of Shor's and Grover's quantum algorithms on these primitives and define AHMC as a post-quantum migration architecture for blockchain transaction authorization, ledger integrity, encrypted off-chain channels, and healthcare-ledger auditability.

Classical Cryptographic Primitives

RSA key generation chooses two large primes p and q and computes $n = p \times q$ Euler's totient is $\varphi(n) = (p-1)(q-1)$. A public exponent e is selected with $\gcd(e, \varphi(n)) = 1$, and the private exponent d is computed such that $d \cdot e \equiv 1 (\mathrm{mod}\, \varphi(n))$. Encryption of message M (represented as an integer modulo) yields ciphertext $C = M^e \bmod n$, and decryption yields. $M = C^d \bmod n$ RSA's security assumption is that factoring n is computationally infeasible at cryptographically relevant sizes.[7,31,55] Elliptic Curve Cryptography (ECC) and ECDSA ECC defines a curve over a finite field, commonly written as $y^2 = x^3 + ax + b (\mathrm{mod}\, p)$, with parameters chosen to avoid singularities. A base point G of large prime order n is selected. The private key is an integer $d (1 \le d < n)$ and the public key is $Q = dG$. In ECDSA, a signer hashes a message m to $H(m)$, samples a per-signature nonce k, computes $R = kG$ and $r = x_R \bmod n$, and then computes $s = k^{-1}(H(m) + d \cdot r) \bmod n$. A verifier computes $w = s^{-1} \bmod n, u_1 = H(m) \cdot w \bmod n, u_2 = r \cdot w \bmod n$, and checks whether the x-coordinate of $u_1 G + u_2 Q$ reduced mod n equals r. The security of ECDSA reduces to the intractability of solving for d given $Q = dG$ (the ECDLP) under the chosen curve parameters.[11,12] Cryptographic hash functions map arbitrary-length inputs m to fixed-length digests $h = \mathrm{hash}(m)$. Desired properties include preimage resistance, second-preimage resistance, and collision resistance. In blockchains, hashes bind transactions into Merkle trees and link blocks by embedding the previous block hash into the next block header.[29,30] For an n-bit digest, generic brute-force preimage search is $O(2^n)$ and generic collision search is $O(2^{n/2})$ under classical assumptions.

Quantum Threat Model

Shor's Algorithm enables polynomial-time integer factorization and discrete logarithms on a sufficiently large, error-corrected quantum computer.[17,18] As a result, RSA security collapses once n can be factored at scale, and ECC/ECDSA collapses once ECDLP instances can be solved for the curves used in production cryptocurrencies.[19] In cryptocurrency contexts, this threatens transaction authorization: if an attacker can derive d from a revealed public key Q, they can generate valid signatures and impersonate the key holder.[27] Grover's Algorithm Grover's algorithm accelerates unstructured search, reducing $O(2^n)$ brute-force search to roughly $O(2^{n/2})$ quantum queries.[21] For symmetric primitives and hash functions, this does not constitute an immediate "break," but it reduces effective security at fixed key or digest sizes, motivating parameter increases for long-term resilience.[22,24]

AHMC Scheme Design Principles

The Advanced Hybrid Module-LWE & Code-Based (AHMC) approach is a defense-in-depth migration architecture that combines multiple post-quantum families with distinct deployment roles. Here, AHMC is explicitly defined as a migration

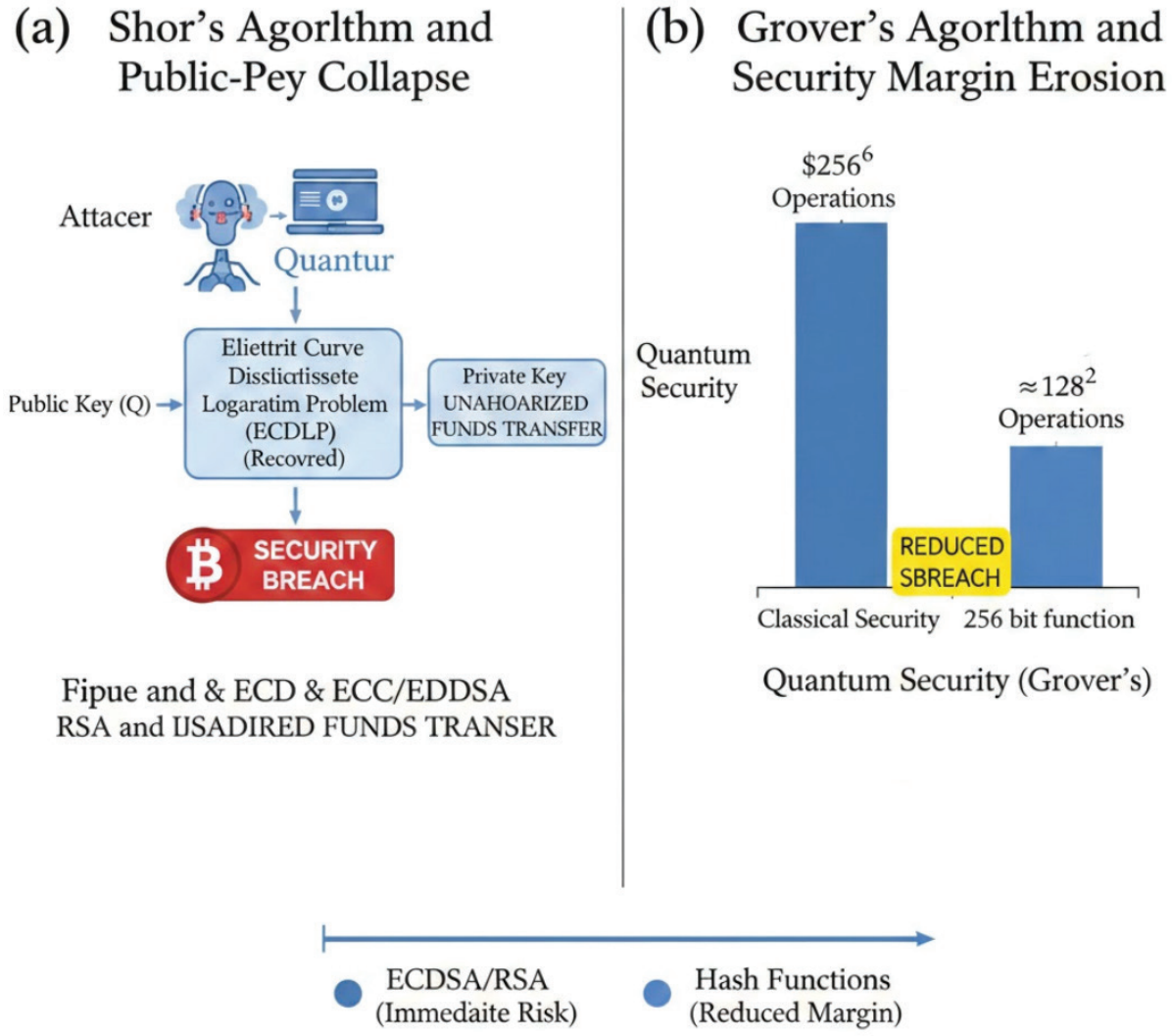

Fig. 1. The Quantum Threat Landscape for Traditional Blockchain Cryptography. (a) Shor's Algorithm and Public-Key Collapse: A visualization of the polynomial-time threat to asymmetric primitives. On a fault-tolerant quantum computer, Shor's algorithm efficiently solves the Elliptic Curve Discrete Logarithm Problem (ECDLP), enabling the recovery of a private key from an exposed public key, leading to unauthorized fund transfers and a total security breach. (b) Grover's Algorithm and Security Margin Erosion: A comparison of the computational work factors for cryptographic hash functions. Under Grover's algorithm, the brute-force search space for a 256-bit function is quadratically reduced from to approximately operations, neessitating increased parameter sizes to maintain long-term security margins.

architecture and protocol profile, not a new cryptographic primitive. Its security should derive from standardized, publicly analyzed components rather than from unreviewed algorithms. For production-oriented blockchain deployment, AHMC binds a transaction to (i) a legacy authorization method such as ECDSA/Schnorr during migration and (ii) a PQC signature selected from FIPS 204 ML-DSA or FIPS 205 SLH-DSA. For key establishment and encrypted off-chain data exchange, AHMC uses FIPS 203 ML-KEM with parameters selected from ML-KEM-512, ML-KEM-768, and ML-KEM-1024. This component is most relevant to wallet-to-wallet encrypted messaging, custody workflows, healthcare off-chain record exchange, and permissioned blockchain membership channels, not to Bitcoin-like transaction signing itself. Signature component. For transaction authentication, AHMC uses ML-DSA as the default deployment profile because it offers shorter signatures than SLH-DSA; SLH-DSA is treated as a conservative hash-based fallback. The previously described code-based signature component is retained only as a research-level diversity option because no code-based digital-signature standard currently occupies the same implementation status as FIPS 204 or FIPS 205. Parameter selection. Table 1 provides concrete AHMC profiles. AHMC-L3 is recommended as the default baseline for serious migration experiments; AHMC-L1 is suitable for testnets and constrained prototypes; AHMC-L5 is appropriate for long-lived, high-assurance healthcare audit logs or high-value custody systems.

Formal Security Model

The security game is a multi-user ledger unforgeability game. The adversary can observe all on-chain data, adaptively request signatures on transactions of its choice, corrupt a bounded set of users, submit candidate transactions to validators, and in the quantum phase, obtain Shor-style breaks of RSA/ECC and Grover-style square-root speedups against generic search. A successful attack is an accepted transaction spending a non-corrupted user's funds or altering a protected healthcare-ledger state without an authorization signature

Citation: Blockchain in Healthcare Today 2026, **9**: 496 - https://doi.org/10.30953/bhty.v9.496

Table 1. Concrete AHMC parameter profiles for blockchain and healthcare ledger deployment.

Profile	Key establishment / encryption profile	Transaction signature profile	Deployment interpretation
AHMC-L1	ML-KEM-512: n = 256, q = 3329, k = 2; ek/dk/ct = 800/1632/768 B; RBG >=128 bits.	ML-DSA-44: pk/sig = 1312/2420 B. Optional SLH-DSA-128s: pk/sig = 32/7856 B.	Testnet or compatibility profile; not preferred for long-retention healthcare records.
AHMC-L3	ML-KEM-768: n = 256, q = 3329, k = 3; ek/dk/ct = 1184/2400/1088 B; RBG ≥ 192 bits.	ML-DSA-65: pk/sig = 1952/3309 B. Optional SLH-DSA-192s: pk/sig = 48/16224 B.	Analytical baseline for pilot benchmarking for blockchain migration experiments and healthcare-ledger pilots.
AHMC-L5	ML-KEM-1024: n = 256, q=3329, k = 4; ek/dk/ct = 1568/3168/1568 B; RBG >= 256 bits.	ML-DSA-87: pk/sig = 2592/4627 B. Optional SLH-DSA-256s: pk/sig = 64/29792 B.	High-assurance profile for long-retention auditability and high-value custody.

ek, encapsulation key; dk, decapsulation key; ct, ciphertext; pk, public key; sig, signature; AHMC, Advanced Hybrid Module-LWE & Code-Based; ML-KEM, Module-Lattice-Based Key-Encapsulation Mechanism Standard; ML-DSA, Module-Lattice-Based Digital Signature Standard; SLH-DSA, Stateless Hash-Based Digital Signature Standard. ML-KEM sizes follow FIPS 203; ML-DSA sizes follow FIPS 204; SLH-DSA sizes follow FIPS 205.[37–39]

produced by that user. AHMC requires existential unforgeability under chosen-message attack (EUF-CMA) for the deployed PQC signature. During hybrid migration, a transaction is valid only if both the legacy signature and the PQC signature verify; under this composition, the hybrid authorization remains unforgeable as long as at least one required signature scheme remains unforgeable for the transaction domain.

For ML-KEM uses, the target is indistinguishability under adaptive chosen-ciphertext attack (IND-CCA2) for encrypted session establishment. Ledger integrity additionally requires domain separation, replay protection, canonical transaction serialization; context strings binding the chain identifier, network, input index, and script state, and key-management rules preventing cross-protocol key reuse.

Implementation Details for Blockchain Integration

Key registration: A migrating wallet generates a PQC signature key pair and publishes either the PQC public key or a commitment to it, depending on the chain's storage model. A permissioned healthcare ledger can register PQC public keys in a certificate or membership service, while a permissionless chain may use script commitments, address-version upgrades, or account-state fields.

Signing: The signer serializes the transaction canonically, applies a domain-separated message hash, and signs the exact authorization message with the legacy key and the PQC key during the hybrid period. In the PQC-only period, the legacy signature is removed, but the transaction format keeps explicit algorithm identifiers and length checks to prevent parsing ambiguity.

Verification: A node or validator checks the key registration, validates canonical encoding, rejects malformed lengths, verifies the legacy signature when required, verifies the PQC signature, and binds the accepted authorization to a single chain and input. For healthcare ledgers, the same verification path applies to consent-state changes, access-log entries, provenance events, and audit records; protected health information should remain off-chain, with only hashes, pointers, commitments, or encrypted metadata placed on-chain.

For a healthcare-ledger deployment, the on-chain evidence for an off-chain EHR object or consent record can be represented as a salted integrity commitment:

$$C_i^{\mathrm{EHR}} = H\left(\rho_i \,\|\, PID_i^* \,\|\, H\left(ptr_i\right) \|\, \mathrm{event}_i \|\, v_i \,\|\, t_i \,\|\, H\left(R_i\right)\right) \quad (H1)$$

where R_i is the off-chain clinical record, consent object, or audit artifact; $H(R_i)$ is its cryptographic digest; PID_i^* is a pseudonymous patient identifier; ptr_i is an off-chain storage pointer or locator; $event_i$ denotes the event type, such as consent update, access event, or provenance update; v_i is the record version; t_i is the timestamp; and ρ_i is a random salt or nonce. Only C_i^{EHR}, metadata, and verification material are stored onchain, while the underlying protected health information remains off-chain.

Statistical methods

No statistical hypothesis tests were performed. Size estimates are deterministic byte counts from published algorithm parameters; verification cost is reported as algorithmic operation counts and expected relative node workload rather than as measured wall-clock time. Empirical benchmarking on production blockchain clients, hardware wallets, and healthcare-ledger testbeds remains future work.

Results

We summarize the quantum-relevant vulnerabilities of RSA, ECC/ECDSA, and cryptographic hash functions as they are used in cryptocurrency protocols, then evaluate AHMC with concrete standards-aligned profiles, transaction-size overhead, verification cost, and comparison against current NIST PQC options.

Quantum Vulnerabilities of RSA and ECC

RSA derives security from the presumed hardness of factoring a large composite modulus n = p×q. Shor's algorithm provides a polynomial-time method for integer factorization on a sufficiently large quantum computer.[17,18] By factoring n to recover p and q, an attacker can compute (n) and derive the private exponent d, enabling decryption of RSA-encrypted material and forgery of RSA-based signatures. While RSA is uncommon for transaction signing in modern cryptocurrencies, RSA remains relevant for some legacy systems and supporting infrastructure; any RSA-dependent component should therefore be treated as non-viable in a post-quantum setting. Modern cryptocurrencies primarily use ECC in the form of ECDSA for digital signatures.[12] ECC's security relies on the hardness of the elliptic curve discrete logarithm problem (ECDLP): given a base point G and a public key Q = dG, it is classically infeasible to recover d. Shor's discrete logarithm algorithm for elliptic curves makes this recovery efficient on a sufficiently powerful quantum computer.[19] Once d is recovered, an attacker can forge valid ECDSA signatures, authorize fraudulent transactions, and undermine blockchain trust assumptions. Shor's algorithm reduces factoring and

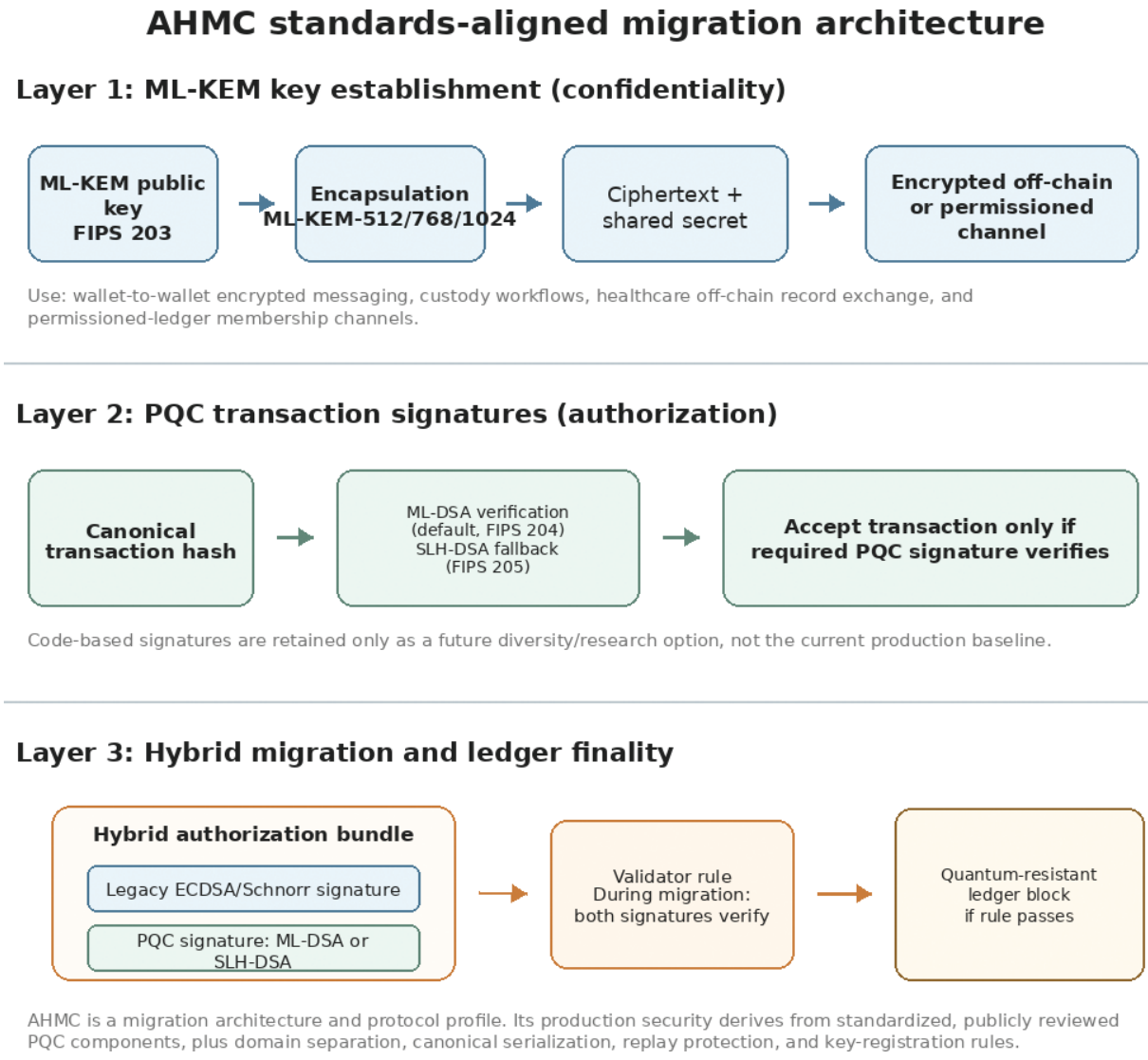

Fig. 2. Architectural Overview of the AHMC Hybrid Framework. This schematic outlines the defense-in-depth strategy of the Advanced Hybrid Module-LWE & Code-Based (AHMC) migration architecture. Layer 1 uses lattice-based Module-Learning With Errors (Module-LWE), instantiated in production profiles through ML-KEM, to support quantum-resistant key establishment for encrypted off-chain channels and permissioned-ledger communication. Layer 2 uses a standards-aligned PQC transaction-signature profile such as ML-DSA by default, SLH-DSA or future code-based alternatives as fallback/diversity options to verify transaction authenticity. Layer 3 illustrates the hybrid migration approach, where a legacy ECDSA signature is combined with a PQC signature during transition. This design is intended to preserve authorization during migration if at least one required signature mechanism remains unforgeable. ML-KEM, Module-Lattice-Based Key-Encapsulation Mechanism Standard; ML-DSA, Module-Lattice-Based Digital Signature Standard; SLH-DSA, Stateless Hash-Based Digital Signature Standard.

discrete logarithms from sub-exponential or exponential classical cost to polynomial-time quantum cost. Key sizes commonly used today would therefore become vulnerable once large-scale, error-corrected quantum hardware becomes available.[20]

Quantum Threats to Hash Functions

Grover's algorithm yields a quadratic speedup for unstructured search, reducing a brute-force space of size 2^n to approximately $2^{n/2}$ quantum queries.[21] For hash functions, this reduces the effective security margin for generic preimage search and motivates stronger parameters for long-term safety. For example, a 256bit hash used for preimage resistance would have its brute-force work factor reduced from roughly 2^{256} to roughly 2^{128} under idealized Grover assumptions. While 2^{128} remains astronomically large, the loss of security margin is material for systems that require multidecade security or that may face highly resourced adversaries. In proof-of-work mining, a quantum miner could in principle use Grover-style search to find a block header satisfying a difficulty target faster than a classical miner, potentially changing the economics of mining and raising centralization risks.[21,23,24,27] The practical impact depends on quantum hardware constraints. Nevertheless, the analysis motivates planning for longer hash outputs, larger security parameters, or other compensating design choices in any long-term post-quantum roadmap.

The AHMC Profile Post-Quantum Resilience

AHMC's post-quantum resilience depends on standardized components and explicit composition rules (Figure 2). ML-KEM is an IND-CCA2 KEM derived from Module-LWE and is used for encrypted key establishment, off-chain channels, and permissioned-ledger communication rather than as a

Citation: Blockchain in Healthcare Today 2026, **9**: 496 - https://doi.org/10.30953/bhty.v9.496

Table 2. Overview of cryptographic techniques in cryptocurrencies.

Cryptographic technique	Key equations / concepts	Mathematical foundation	Used in
RSA encryption	$n = pq$, where p, q are large primes. $\varphi(n) = (p-1)(q-1)$. Choose e such that $\gcd(e, \varphi(n)) = 1$. Find d such that $ed \equiv 1 \pmod{\varphi(n)}$. Encryption: $c \equiv m^e \pmod{n}$. Decryption: $m \equiv c^d \pmod{n}$.	Integer factorization; modular arithmetic; multiplicative group $\mathbb{Z}_n^*$	Key management; legacy encryption and signature schemes
Elliptic curve cryptography	Curve over finite field $\mathbb{F}_p$: $y^2 \equiv x^3 + ax + b \pmod{p}$, with $4a^3 + 27b^2 \neq 0 \pmod{p}$. Scalar multiplication: $Q = kG$.	Elliptic Curve Discrete Logarithm Problem (ECDLP); finite field arithmetic; elliptic-curve group operations.	Public-key cryptography; basis of ECDSA signatures such as Bitcoin, Ethereum.
ECDSA (digital signatures)	Private key $d \in [1, n-1]$. Public key $Q = dG$. Choose nonce k. $R = kG = (x_R, y_R)$. $r = x_R \bmod n$ $s = k^{-1}(H(m) + dr) \bmod n$. Verify using $u_1 = H(m)s^{-1} \bmod n, u_2 = rs^{-1} \bmod n$; valid if $r \equiv (u_1G + u_2Q)_x \bmod n$.	ECDLP; elliptic-curve group structure; modular inversion; hash functions.	Transaction authentication in blockchain systems.
Hash functions (SHA-256, Keccak-256)	$H: \{0,1\}^* \longrightarrow \{0,1\}^n$. $h = H(m)$. Merkle tree parent: $H(h_L \parallel h_R)$.	Preimage resistance; second-preimage resistance; collision resistance; compression-function or sponge construction.	Transaction hashing; Merkle trees; address derivation; proof-of-work.

Equations are shown in simplified form for readability; see Methods for definitions and context. ECDSA: elliptic curve digital signature algorithm; RSA: Rivest–Shamir–Adleman; SHA-256: Secure Hash Algorithm 256-bit.

Table 3. Quantum threats to cryptographic protocols.

Cryptographic technique	Threatening quantum algorithm	Mechanism of attack	Impact on cryptocurrencies
RSA	Shor's algorithm	Efficiently factors $n = pq$ in polynomial time. Once p and q are recovered, compute $\varphi(n)$ and derive private key d from $ed \equiv 1 \pmod{\varphi(n)}$.	Breaks RSA confidentiality and enables signature forgery wherever RSA is deployed.
ECC (ECDSA)	Shor's algorithm	Solves the elliptic curve discrete logarithm problem. Given $Q = dG$, recovers private key d.	Enables forged ECDSA signatures and unauthorized transactions once public keys are exposed.
Hash functions	Grover's algorithm	Provides quadratic speedup for brute-force search. Reduces effective preimage resistance from 2^n to approximately $2^{n/2}$.	Reduces security margin; may advantage quantum-capable miners; motivates longer hash outputs.

Grover's impact is idealized; real-world feasibility depends on quantum hardware constraints and implementation details. ECC: Elliptic Curve Cryptography; ECDSA: elliptic curve digital signature algorithm; RSA: Rivest–Shamir–Adleman; SHA-256: Secure Hash Algorithm 256-bit.

transaction-signature mechanism.[37] ML-DSA and SLH-DSA provide NIST-standardized transaction-signature options.[38,39] The code-based element is not claimed as production-ready in this manuscript; it is a future diversification path requiring independent standardization, cryptanalysis, implementation review, and benchmarking. Table 2 summarizes classical cryptographic techniques used in cryptocurrencies, Table 3 summarizes the corresponding quantum threats, and Table 4 compares PQC transaction overheads.

Analytical Benchmarking, Transaction-Size Impact, and Comparison against Current PQC

The key deployment cost for blockchain is not only cryptanalytic security but also the number of bytes placed in each transaction and the number of verifications each node must perform. Because this manuscript does not include a production client implementation, the benchmark below is analytical: it uses byte sizes from FIPS 203, FIPS 204, and FIPS 205 and counts verification operations required by the proposed validation rules.[37–39]

Discussion

Interpretation of Findings

The analysis indicates that quantum computing poses a direct and potentially existential risk to the public-key cryptography that underpins transaction authorization in cryptocurrency ecosystems. In the classical security model, RSA and elliptic-curve

Table 4. Transaction-size and verification-cost comparison against current PQC standards.

Scheme/profile	Public key and signature/ciphertext bytes	Verification or operation cost	Transaction-size impact
ECDSA/secp256k1 baseline	Pk = 33 B compressed or 65 B uncompressed; signature commonly 64–72 B depending encoding.	One elliptic-curve signature verification per input or authorization.	Baseline for Bitcoin-like transactions.
ML-DSA-44	pk = 1312 B; sig = 2420 B.	One ML-DSA verification; no ECDLP assumption.	+approx. 2.35 KB per signature versus a 72 B ECDSA signature.
ML-DSA-65	Pk = 1952 B; sig = 3309 B.	One ML-DSA verification; recommended AHMC-L3 baseline.	+approx. 3.24 KB per signature versus a 72 B ECDSA signature.
ML-DSA-87	pk = 2592 B; sig = 4627 B.	One ML-DSA verification at higher security level.	+approx. 4.56 KB per signature versus a 72 B ECDSA signature.
SLH-DSA small profiles	pk = 32/48/64 B; sig = 7856/16224/29792 B for 128s/192s/256s.	Hash-based verification; conservative assumption diversity but high bandwidth.	+approx. 7.8 KB to 29.7 KB per signature before encoding overhead.
AHMC hybrid L3	ECDSA signature + ML-DSA-65 signature approx. 3.38 KB, plus PQC public-key registration where required.	One ECDSA verification and one ML-DSA verification during migration.	For m inputs, overhead is roughly m × 3.3 KB plus key-registration data; PQC-only removes the legacy signature.
ML-KEM (off-chain use)	Ek = 800/1184/1568 B; ct = 768/1088/1568 B for ML-KEM-512/768/1024.	Encapsulation and decapsulation, not transaction-signature verification.	No direct replacement for ECDSA; useful for encrypted channels, custody workflows, and healthcare off-chain data exchange.

Values exclude chain-specific serialization, witness/script, account-state, and public-key-registration overhead. For Bitcoin-like UTXO (unspent transaction output) systems, the dominant cost scales with the number of signed inputs. For account-based or permissioned healthcare ledgers, PQC public keys can often be registered once and referenced by identifier, while signatures are attached to state transitions, consent updates, or audit events. AHMC, Advanced Hybrid Module-LWE & Code-Based; ML-KEM, Module-Lattice-Based Key-Encapsulation Mechanism Standard; ML-DSA, Module-Lattice-Based Digital Signature Standard; SLH-DSA, Stateless Hash-Based Digital Signature Standard.

cryptography, including ECDSA are trusted because the underlying computational problems, integer factorization for RSA and discrete logarithms for ECC, are believed to be infeasible at cryptographically relevant key sizes with conventional hardware.

Quantum computing alters that premise: Shor's algorithm provides an efficient route to solving both integer factorization and discrete logarithms once sufficiently capable, fault-tolerant quantum computers become available, collapsing the hardness assumptions that make RSA and ECC practical for authentication and key ownership proofs.[17–19] In other words, quantum advantage is not merely a marginal speed improvement; it is a structural change in complexity class that would transform today's "effectively impossible" attacks into attacks that are plausible for a determined, well-resourced adversary given adequate quantum resources.

In practical cryptocurrency deployments, this risk concentrates most sharply in digital signatures because signatures are the primary mechanism used to authorize transfers and prevent impersonation. The ledger's integrity depends on the assumption that only the legitimate key holder can produce a valid signature over a transaction that spends funds. If a public key is exposed in a way that allows an attacker to target it and a quantum adversary can recover the corresponding private key, the attacker can forge signatures, authorize unauthorized transfers, and defeat the authenticity guarantees that support the economic meaning of the ledger.[27] This is not merely a localized threat to a single wallet or a small subset of users. If quantum-enabled key recovery becomes feasible at scale, the resulting signature forgeries would undermine systemic trust: the network could no longer reliably distinguish legitimate authorization from adversarial impersonation, and the credibility of balances and settlement finality would erode. Even the perception that such attacks are imminent can have destabilizing effects, because cryptocurrency security is inseparable from user confidence in the immutability and enforceability of ownership.

The operational implications extend beyond "eventual break" scenarios. Cryptocurrencies are long-lived systems: addresses, unspent outputs, and keys may remain relevant for years, and the cost of migrating cryptographic primitives is high because it must be coordinated across wallets, exchanges, nodes, miners/validators, and governance processes. The analysis therefore supports a risk-management interpretation rather than a purely technical one: the relevant question is not only whether large-scale quantum computers exist today, but whether ecosystems can credibly transition before quantum capability reaches a threshold that creates actionable exploitation risk. From that perspective, delay is itself a vulnerability, because a rushed upgrade under time pressure raises the probability of design mistakes, contentious protocol changes, and fragmented adoption. Accordingly, the primary finding is that public-key components should be treated as "on the clock," and that the window for orderly migration is determined by engineering and governance lead time, not only by hardware timelines. Hash functions, by contrast, are not "broken" in the same absolute sense by quantum computing, but they do experience a meaningful reduction in effective security margin under Grover's algorithm. Grover's algorithm provides a quadratic speedup for generic unstructured search, which translates into a square-root reduction in brute-force work factors for preimage-style attacks at fixed output lengths.[21] This does not automatically imply that hashes become unusable; rather, it changes the security calculus for systems designed to maintain strong security for multi-decade horizons. A conservative

Citation: Blockchain in Healthcare Today 2026, **9**: 496 - https://doi.org/10.30953/bhty.v9.496

interpretation is that hash-based security should not be assumed to be permanently "future-proof" without parameter adjustments. If a design depends on a specific work factor remaining out of reach even for a highly resourced adversary, then the Grover-induced margin loss should be explicitly addressed through longer outputs, stronger parameterization, or architectural choices that preserve the intended security level. This margin erosion also has protocol-level and economic implications. Hash functions are embedded not only in block linking and data integrity, but also in proof-of-work mining and other consensus-related mechanisms.

In proof-of-work settings, Grover-style speedups could, in principle, advantage quantum-capable miners by reducing the expected search effort required to find a valid block header under a given difficulty target, with downstream effects on decentralization and security economics.[21,23,24,27] The feasibility and magnitude of any such advantage depend on practical constraints, quantum circuit depth, error correction overhead, and the extent to which the search can be parallelized without eroding the theoretical benefit, so the conclusion is not that quantum mining dominance is inevitable. Rather, the key interpretation is that "hash-only reliance" is not automatically safe as a long-term strategy: the ecosystem must consider how even partial quantum advantages could alter incentives, concentrate power, or change security assumptions. Taken together, these findings motivate a proactive and staged approach to quantum risk. The public-key signature layer represents the highest-impact single point of failure because it directly governs spending authorization and identity binding in permissionless ledgers. Hash mechanisms retain significant resilience but warrant parameter planning to maintain intended safety margins under Grover-style reductions. The overarching interpretation is therefore twofold: (1) quantum risk is asymmetric across cryptographic components, and mitigation should prioritize signature/identity primitives; and (2) the cost of migration argues for early planning, because security in decentralized systems is not only a matter of mathematical hardness but also of deployability, coordination, and sustained trust.

Healthcare Translation of the Results and Implementation Priorities

The healthcare relevance of this analysis arises from a shared dependence on long-lived digital signatures and verifiable ledger events, not from an assumption that cryptocurrency and healthcare are operationally equivalent. Bitcoin and other public cryptocurrencies provide mature, adversarially tested environments in which public-key exposure, signature forgery, transaction-size pressure, and validator workload can be analyzed explicitly. Healthcare ledgers, by contrast, are generally identity-bound, permissioned, and integrated with clinical, administrative, and research systems. Their principal concerns include confidentiality, accountable identity, availability, revocation, audit retention, and continuity of care. Nevertheless, both environments depend on the assumption that only an authorized key holder can create a valid signature. If a future quantum adversary can recover a private key from an exposed RSA- or ECC-based public key, a healthcare organization may no longer be able to determine reliably whether a consent decision, identity assertion, access event, provenance record, or audit entry was authorized by the legitimate patient, provider, institution, device, or custodian.[17–19,53,54]

The healthcare applications most directly connected to the present results are those in which a blockchain or distributed ledger records durable evidence about an event rather than the underlying clinical information itself. Examples include patient consent grants and revocations, provider and medical-device credential status, access-log anchoring, clinical-trial protocol and data-provenance events, pharmaceutical supply-chain attestations, and integrity commitments associated with off-chain electronic health record objects.[53,54] In these applications, the post-quantum objective is durable authenticity. A future verifier should be able to establish that the recorded state transition was authorized by the appropriate actor and that the related off-chain object has not been altered. Protected health information should therefore remain within governed off-chain repositories, while the ledger contains only the minimum verification material required for auditability, such as hashes, salted commitments, pseudonymous identifiers, pointers, event types, timestamps, key identifiers, and signatures.

AHMC maps to these healthcare workflows through a separation of cryptographic functions. Within the AHMC architecture, ML-KEM supports quantum-resistant key establishment for encrypted off-chain exchange, institutional communication channels, custody workflows, and permissioned-ledger membership services. ML-KEM is not used as a substitute for a digital signature. ML-DSA or SLH-DSA provides post-quantum authorization for consent changes, credential updates, provenance attestations, and other signed ledger events. During the migration period, AHMC can require both the existing ECDSA or Schnorr signature and the applicable post-quantum signature. This hybrid rule preserves continuity with established infrastructure while requiring a quantum-resistant authorization path. The two signatures should use independent keys, cover the same canonical and domain-separated authorization message, and be verified under rules that reject ambiguous encodings, replayed events, algorithm-downgrade attempts, and partially validated authorization bundles.[37–39]

The analytical transaction-size results have direct implications for healthcare deployment. The ML-DSA-44, ML-DSA-65, and ML-DSA-87 signatures evaluated in this study are approximately 2.4 KB, 3.3 KB, and 4.6 KB, respectively, before ledger-specific serialization and encoding overhead. The AHMC hybrid L3 configuration requires approximately 3.38 KB for the combined ECDSA and ML-DSA-65 signatures, in addition to any public-key-registration data. These sizes are considerably larger than conventional ECDSA signatures. For healthcare systems, the relevant question is therefore not simply whether a post-quantum algorithm is secure, but whether the resulting storage, propagation, verification, backup, and recovery costs are acceptable for the intended workflow. A low-volume consent ledger, clinical-trial provenance system, or institutional credential registry may tolerate overhead that would be unsuitable for a high-volume clinical-message stream.

Healthcare implementations should consequently apply AHMC according to the value and retention requirements of the event being protected. It is not necessary or desirable to place every clinical message on a blockchain or attach a

post-quantum ledger signature to every exchange of healthcare data. AHMC is most relevant to high-consequence state transitions whose authenticity may be questioned years after their creation. These can include a patient's grant or withdrawal of consent, the issuance or revocation of an institutional credential, an attestation concerning the origin of research data, or an integrity commitment linked to an audit artifact. Permissioned healthcare ledgers may reduce repeated storage costs by registering public keys once and referring to them through stable identifiers. They may also batch related events or store large signatures in an associated audit repository, provided that these optimizations preserve independent verification and do not make future validation dependent on an unavailable proprietary service.

A healthcare migration should begin with a cryptographic inventory rather than an immediate system-wide replacement. Organizations should identify where RSA and ECC are used across ledger nodes, identity and certificate services, application programming interfaces, hardware security modules, medical devices, institutional gateways, and archival-verification processes. Signed records should then be classified according to their required trust lifetime and the consequence of forgery. Consent, identity, provenance, and audit records that may remain legally, clinically, or scientifically consequential for many years should receive priority. A controlled AHMC pilot can then evaluate hybrid key registration and signing for a limited group of high-value state transitions before a broader transition is attempted.

Long-term validation is especially important in healthcare because the evidentiary value of an event may extend beyond the active lifetime of the original key, certificate, software platform, or ledger implementation. An AHMC deployment should retain the algorithm identifier, applicable public-key-registration history, signature policy, event context, timestamp, record version, and relevant revocation information required to verify historical entries. Key rotation should not make previous records unverifiable. Healthcare organizations should also define procedures for compromised keys, unavailable signers, emergency access, re-signing or archival protection of older evidence, and migration from hybrid authorization to post-quantum-only authorization. These controls should be established before the original cryptographic mechanisms become operationally unsafe rather than during an emergency cutover.[37–39,48]

AHMC alone does not establish healthcare security, privacy, or regulatory compliance. Its cryptographic protections must operate together with identity proofing, role- and attribute-based access controls, least-privilege authorization, key custody, revocation, data minimization, secure time-stamping, audit review, software-supply-chain controls, incident response, and continuity procedures. Implementations must also address practical vulnerabilities such as weak randomness, parsing errors, timing leakage, cache leakage, power analysis, fault injection, replay, and downgrade attacks.[56,57] The AHMC profiles evaluated here should therefore be understood as candidates for controlled technical and workflow evaluation rather than as claims of immediate clinical readiness.

The healthcare contribution of the present analysis is consequently a migration decision framework. It identifies the types of long-lived healthcare assertions that could become unreliable if classical public-key cryptography is broken; maps the distinct AHMC components to authorization and encrypted communication functions; quantifies the signature-size and verification consequences of migration; and identifies the operational controls required to preserve confidentiality, availability, accountable identity, and long-term auditability. Cryptocurrency provides a demanding benchmark for cryptographic overhead and adversarial exposure, while healthcare determines whether the same migration architecture can support durable trust without disrupting clinical and administrative operations.

Limitations

This innovative work remains implementation-oriented rather than a completely deployed protocol. AHMC is not a new cryptographic primitive; it is a conceptual migration architecture that must be instantiated with standardized components for production. The code-based component is not recommended for production transaction signing until a concrete standardized code-based signature or related mechanism is selected, cryptanalyzed, implemented, and benchmarked. The cost model uses published byte sizes and verification steps, but it does not include empirical node benchmarks, mempool fee simulations, hardware-wallet measurements, or side-channel testing. Real-world quantum attack feasibility depends on quantum hardware scale, error correction, and operational cost, and these factors remain uncertain. Implementation security is also critical: even mathematically sound PQC schemes can be compromised by timing, cache, power, fault-injection, or randomness failures if engineering is insufficient.[56,57]

Future Directions

Future work should prioritize (1) producing open, audited AHMC-L3 reference implementations for wallets, nodes, hardware wallets, and permissioned healthcare ledgers; (2) benchmarking key generation, signing, verification, transaction propagation, block validation, storage growth, and fee impact under ML-DSA-44/65/87 and SLH-DSA alternatives; (3) formally verifying hybrid authorization rules, domain separation, replay protection, and key-registration logic; (4) comparing AHMC against direct ML-DSA-only, SLH-DSA-only, and ECDSA+ML-DSA migration paths on public testnets; and (5) engineering side-channel-resistant implementations with constant-time parsing, robust randomness, secure key erasure, and hardware-wallet support. Healthcare pilots should additionally evaluate consent-update latency, audit-log verification, off-chain encrypted record exchange, and regulatory requirements for long-term signature validation.

Conclusions

Quantum computing creates a long-term threat to the RSA- and ECC-based signatures used to authorize transactions and preserve trust in distributed ledgers.[17–19] For healthcare, the relevant risk is not cryptocurrency price or mining performance, but the future verifiability of consent changes, access decisions, identity credentials, provenance events, and audit evidence that may remain consequential for many years. This review uses Bitcoin and cryptocurrency systems as an adversarial benchmark for cryptographic failure and deployment

Citation: Blockchain in Healthcare Today 2026, **9**: 496 - https://doi.org/10.30953/bhty.v9.496

overhead, then translates the findings into a standards-aligned hybrid PQC migration profile based on FIPS 203 ML-KEM, FIPS 204 ML-DSA, and FIPS 205 SLH-DSA.[37–39]

The AHMC label denotes this migration profile only; it is not a proprietary algorithm or a new cryptographic primitive. Healthcare adoption should proceed through cryptographic inventory, risk-based prioritization, limited hybrid pilots, empirical performance testing, key-lifecycle governance, and side-channel-resistant implementation, with protected health information retained off-chain. These steps provide a practical basis for preserving long-term authenticity and auditability without conflating permissionless cryptocurrency deployment with healthcare operations.

Funding

This study received no external funding. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript. All authors declare no financial or non-financial relationships or activities that could bias the interpretation of the work.

Financial and Non-Financial Relationship and Activities

Not applicable.

Data Availability Statement (DAS), Data Sharing, Reproducibility, and Data Repositories

No data were generated or analyzed in the presented research. Theoretical derivations and supporting notes are available from the authors upon reasonable request.

Application of AI-Generated Text or Related Technology

No generative AI images were created or used in this manuscript.

Contributions

RK conceived the study, performed the cryptographic and quantum-threat analysis, and drafted the manuscript. MK contributed to the post-quantum design discussion and edited the manuscript for technical accuracy. DR contributed to the blockchain integration considerations and revised the manuscript critically. All authors approved the final version and agree to be accountable for all aspects of the work.

Acknowledgments

The authors acknowledge the broader research community in post-quantum cryptography and quantum resource estimation whose published work informs this review. No additional contributors who do not meet authorship criteria are listed.

References

1. Nakamoto S. Bitcoin: a peer-to-peer electronic cash system [Internet]. 2008 [cited 2026 Jan 23]. Available from: https://bitcoin.org/bitcoin.pdf
2. Antonopoulos AM. Mastering bitcoin: unlocking digital cryptocurrencies. 2nd ed. Sebastopol, CA: O'Reilly Media; 2017.
3. Narayanan A, Bonneau J, Felten E, Miller A, Goldfeder S. Bitcoin and cryptocurrency technologies: a comprehensive introduction. Princeton, NJ: Princeton University Press; 2016.
4. Buterin V. Ethereum whitepaper: a next-generation smart contract and decentralized application platform [Internet]. 2014 [cited 2026 Jan 23]. Available from: https://ethereum.org/whitepaper/
5. Buterin V, Griffith V. Casper the friendly finality gadget [Preprint]. 2017 [cited 2026 Jan 23]. Available from: https://arxiv.org/abs/1710.09437
6. Diffie W, Hellman M. New directions in cryptography. IEEE Trans Inf Theory. 1976;22(6):644–54. https://doi.org/10.1109/TIT.1976.1055638
7. Rivest RL, Shamir A, Adleman L. A method for obtaining digital signatures and public-key cryptosystems. Commun ACM. 1978;21(2):120–6. https://doi.org/10.1145/359340.359342
8. Miller VS. Use of elliptic curves in cryptography. In: Williams HC, editor. Advances in cryptology, CRYPTO '85. Lecture Notes in Computer Science. Berlin: Springer; 1986. p. 417–26.
9. Koblitz N. Elliptic curve cryptosystems. Math Comput. 1987;48(177):203–9. https://doi.org/10.1090/S0025-5718-1987-0866109-5
10. Boneh D, Shoup V. A graduate course in applied cryptography [Internet]. 2020 [cited 2026 Jan 23]. Available from: https://toc.cryptobook.us/
11. Hankerson D, Menezes A, Vanstone S. Guide to elliptic curve cryptography. New York, NY: Springer; 2004.
12. Johnson D, Menezes A, Vanstone S. The elliptic curve digital signature algorithm (ECDSA). Int J Inf Secur. 2001;1(1):36–63. https://doi.org/10.1007/s102070100002
13. Nielsen MA, Chuang IL. Quantum computation and quantum information. 10th anniversary ed. Cambridge: Cambridge University Press; 2010.
14. Kaye P, Laflamme R, Mosca M. An introduction to quantum computing. Oxford: Oxford University Press; 2007.
15. Mavroeidis V, Vishi K, Zych MD, Jøsang A. The impact of quantum computing on present cryptography. Int J Adv Comput Sci Appl. 2018;9(3):405–14. https://doi.org/10.14569/IJACSA.2018.090354
16. National Academies of Sciences, Engineering, and Medicine. Quantum computing: progress and prospects. Washington, DC: The National Academies Press; 2019.
17. Shor PW. Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings of the 35th Annual Symposium on Foundations of Computer Science; 1994 Nov 20–22; Santa Fe, NM. Los Alamitos, CA: IEEE Computer Society Press; 1994. p. 124–34. https://doi.org/10.1109/SFCS.1994.365700
18. Shor PW. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J Comput. 1997;26(5):1484–509. https://doi.org/10.1137/S0097539795293172
19. Proos J, Zalka C. Shor's discrete logarithm quantum algorithm for elliptic curves. Quantum Inf Comput. 2003;3(4):317–44. https://doi.org/10.26421/QIC3.4-3
20. Roetteler M, Naehrig M, Svore KM, Lauter K. Quantum resource estimates for computing elliptic curve discrete logarithms [Preprint]. 2017 [cited 2026 Jan 23]. Available from: https://arxiv.org/abs/1706.06752
21. Grover LK. A fast quantum mechanical algorithm for database search. In: Proceedings of the 28th Annual ACM Symposium on Theory of Computing; 1996 May 22–24; Philadelphia, PA. New York, NY: Association for Computing Machinery; 1996. p. 212–19. https://doi.org/10.1145/237814.237866
22. Brassard G, Høyer P, Tapp A. Quantum cryptanalysis of hash and claw-free functions. In: Lucchesi CL, Moura A, editors. LATIN '98: theoretical informatics. Lecture Notes in Computer Science. Berlin: Springer; 1998. p. 163–9.
23. Bernstein DJ. Cost analysis of hash collisions: will quantum computers make SHARCS obsolete? [Internet]. In: SHARCS '09 Workshop Record: Proceedings of the 4th Workshop on Special-purpose Hardware for Attacking Cryptographic Systems; 2009 Sep 9–10; Lausanne; 2009. p. 105–16 [cited 2026 Jan 23]. Available from: https://cr.yp.to/hash/collisioncost-20090517.pdf
24. Amy M, Di Matteo O, Gheorghiu V, Mosca M, Parent A, Schanck JM. Estimating the cost of generic quantum pre-image attacks on SHA-2 and SHA-3 [Preprint]. 2016 [cited 2026 Jan 23]. Available from: https://arxiv.org/abs/1603.09383
25. Bernstein DJ, Lange T. Post-quantum cryptography. Nature. 2017;549(7671):188–94. https://doi.org/10.1038/nature23461
26. Mosca M. Cybersecurity in an era with quantum computers: will we be ready? [Preprint]. Cryptology ePrint Archive. 2015; Report No.:

2015/1075 [cited 2026 Jan 23]. Available from: https://eprint.iacr.org/2015/1075
27. Aggarwal D, Brennen GK, Lee T, Santha M, Tomamichel M. Quantum attacks on bitcoin, and how to protect against them. Ledger. 2018;3:68–90. https://doi.org/10.5195/ledger.2018.127
28. Stewart I, Ilani S, Ozawa T, Yamamoto L. Committing to quantum resistance: a slow defence for bitcoin against a fast quantum computing attack. R Soc Open Sci. 2018;5(6):180410. https://doi.org/10.1098/rsos.180410
29. National Institute of Standards and Technology. SHA-3 standard: permutation-based hash and extendable-output functions [Internet]. Gaithersburg, MD: National Institute of Standards and Technology (US); 2015. (FIPS PUB 202) [cited 2026 Jan 23]. Available from: https://csrc.nist.gov/pubs/fips/202/final
30. Merkle RC. A digital signature based on a conventional encryption function. In: Pomerance C, editor. Advances in cryptology, CRYPTO '87. Lecture Notes in Computer Science. Berlin: Springer; 1988. p. 369–78.
31. Boneh D. Twenty years of attacks on the RSA cryptosystem. Notices Am Math Soc. 1999;46(2):203–13.
32. Bernstein DJ. Introduction to post-quantum cryptography. In: Bernstein DJ, Buchmann J, Dahmen E, editors. Post-quantum cryptography. Berlin: Springer; 2009. p. 1–14.
33. Chen L, Chen L, Jordan S, Liu YK, Moody D, Peralta R, et al. Report on post-quantum cryptography [Internet]. Gaithersburg, MD: National Institute of Standards and Technology (US); 2016. (NISTIR 8105) [cited 2026 Jan 23]. Available from: https://csrc.nist.gov/pubs/ir/8105/final
34. Alagic G, Alperin-Sheriff J, Apon D, Cooper D, Dang Q, Kelsey J, et al. Status report on the second round of the NIST post-quantum cryptography standardization process [Internet]. Gaithersburg, MD: National Institute of Standards and Technology (US); 2020. (NISTIR 8309) [cited 2026 Jan 23]. Available from: https://csrc.nist.gov/pubs/ir/8309/final
35. National Institute of Standards and Technology. Post-quantum cryptography standardization [Internet]. Gaithersburg, MD: National Institute of Standards and Technology (US); 2016 [cited 2026 Jan 23]. Available from: https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization
36. Moody D. The NIST post-quantum cryptography standardization process [Internet]. Gaithersburg, MD: National Institute of Standards and Technology (US); 2020 [cited 2026 Jan 23]. Available from: https://csrc.nist.gov/projects/post-quantum-cryptography
37. National Institute of Standards and Technology. Module-lattice-based key-encapsulation mechanism standard [Internet]. Washington, DC: U.S. Department of Commerce; 2024. (FIPS 203) [cited 2026 Jan 23]. Available from: https://csrc.nist.gov/pubs/fips/203/final
38. National Institute of Standards and Technology. Module-lattice-based digital signature standard [Internet]. Washington, DC: U.S. Department of Commerce; 2024. (FIPS 204) [cited 2026 Jan 23]. Available from: https://csrc.nist.gov/pubs/fips/204/final
39. National Institute of Standards and Technology. Stateless hash-based digital signature standard [Internet]. Washington, DC: U.S. Department of Commerce; 2024. (FIPS 205) [cited 2026 Jan 23]. Available from: https://csrc.nist.gov/pubs/fips/205/final
40. Peikert C. A decade of lattice cryptography. Found Trends Theor Comput Sci. 2016;10(4):283–424. https://doi.org/10.1561/0400000074
41. Micciancio D, Regev O. Lattice-based cryptography. In: Bernstein DJ, Buchmann J, Dahmen E, editors. Post-quantum cryptography. Berlin: Springer; 2009. p. 147–91.
42. Langlois A, Stehlé D. Worst-case to average-case reductions for module lattices. Des Codes Cryptogr. 2015;75(3):565–99. https://doi.org/10.1007/s10623-014-9938-4
43. Stebila D, Mosca M. Post-quantum key exchange for the Internet and the Open Quantum Safe project. In: Avanzi R, Heys H, editors. Selected areas in cryptography – SAC 2016. Lecture Notes in Computer Science. Cham: Springer; 2017. Vol. 10532. p. 1–24.
44. Crockett E, Paquin C, Stebila D. Prototyping post-quantum and hybrid key exchange and authentication in TLS and SSH [Internet]. In: Proceedings of the 2nd NIST Post-Quantum Cryptography Standardization Conference; 2019 Aug 22–24; Santa Barbara, CA. Gaithersburg, MD: National Institute of Standards and Technology (US); 2019 [cited 2026 Jan 23]. Available from: https://csrc.nist.gov/CSRC/media/Events/Second-PQC-Standardization-Conference/documents/accepted-papers/stebila-prototyping-post-quantum.pdf
45. Alkim E, Ducas L, Pöppelmann T, Schwabe P. Post-quantum key exchange: a new hope [Internet]. In: 25th USENIX Security Symposium (USENIX Security 16); 2016 Aug 10–12; Austin, TX. Berkeley, CA: USENIX Association; 2016. p. 327–43 [cited 2026 Jan 23]. Available from: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/alkim
46. Buchmann J, Dahmen E, Szydlo M. Hash-based digital signature schemes. In: Bernstein DJ, Buchmann J, Dahmen E, editors. Post-quantum cryptography. Berlin: Springer; 2009. p. 35–93.
47. Güneysu T, Oder T, Pöppelmann T, Schwabe P. Lattice-based signatures: optimization and implementation on reconfigurable hardware. IEEE Trans Comput. 2015;64(7):1954–67. https://doi.org/10.1109/TC.2014.2346177
48. Bindel N, Brendel J, Fischlin M, Gonçalves B, Steinwandt R. Transitioning to a quantum-resistant public key infrastructure. In: Mosca M, editor. Post-quantum cryptography. Berlin: Springer; 2017. p. 384–405.
49. Sendrier N. Code-based cryptography. In: van Tilborg HCA, Jajodia S, editors. Encyclopedia of cryptography and security. Boston, MA: Springer; 2011.
50. Overbeck R, Sendrier N. Code-based cryptography. In: Bernstein DJ, Buchmann J, Dahmen E, editors. Post-quantum cryptography. Berlin: Springer; 2009. p. 95–145.
51. Ducas L, Durmus A, Lepoint T, Lyubashevsky V. Learning a zonotope and more: cryptanalysis of NTRUSign countermeasures. In: Takagi T, Peyrin T, editors. Advances in cryptology, ASIACRYPT 2017. Lecture Notes in Computer Science. Cham: Springer; 2017. p. 433–63.
52. McEliece RJ. A public-key cryptosystem based on algebraic coding theory [Internet]. Pasadena, CA: Jet Propulsion Laboratory; 1978. (DSN Progress Report 42-44). p. 114–16 [cited 2026 Jan 23]. Available from: https://tda.jpl.nasa.gov/progress_report/42-44/44N.PDF
53. Agbo CC, Mahmoud QH, Eklund JM. Blockchain technology in healthcare: a systematic review. Healthcare (Basel). 2019;7(2):56. https://doi.org/10.3390/healthcare7020056
54. Elangovan D, Long CS, Bakrin FS, Tan CS, Goh KW, Yeoh SF, et al. The use of blockchain technology in the health care sector: systematic review. JMIR Med Inform. 2022;10(1):e17278. https://doi.org/10.2196/17278
55. Maurer UM. Fast generation of prime numbers and secure public-key cryptographic parameters. J Cryptol. 1995;8(3):123–55. https://doi.org/10.1007/BF00202269
56. Kocher P, Jaffe J, Jun B. Differential power analysis. In: Wiener M, editor. Advances in cryptology, CRYPTO '99. Lecture Notes in Computer Science. Berlin: Springer; 1999. p. 388–97.
57. Genkin D, Pachmanov L, Pipman I, Tromer E. Physical key extraction attacks on PCs. Commun ACM. 2016;59(6):70–9. https://doi.org/10.1145/2851486

Copyright Ownership: This is an open-access article distributed in accordance with the Creative Commons Attribution Non-Commercial (CC BY-NC 4.0) license, which permits others to distribute, adapt, enhance this work non-commercially, and license their derivative works on different terms, provided the original work is properly cited and the use is non-commercial. See http://creativecommons.org/licenses/by-nc/4.0. The authors of this article own the copyright.