
Migration to Quantum Safe Blockchains: A Compact Architecture Using MPPK Key Encapsulation Mechanism and HPPK Digital Signatures

Michel Barbeau^{1,*} and Randy Kuang²

¹*Carleton University, Ottawa, Ontario, Canada*

²*Quantropi Inc., Ottawa, Ontario, Canada*

E-mail: michelbarbeau@cunet.carleton.ca; randy.kuang@quantropi.com

**Corresponding Author*

Received 30 July 2026; Accepted 20 August 2026

Abstract

The advent of quantum computing threatens the public-key cryptography that underlies modern blockchains, including ECDSA, Ed25519, and ECDH. Although NIST has standardized lattice-based schemes (ML-KEM and ML-DSA) as quantum-safe standards, their large key and signature sizes (several kilobytes each) pose scalability challenges for blockchains. We propose a quantum-safe blockchain architecture based entirely on multivariate polynomial cryptography, specifically the MPPK KEM key encapsulation mechanism and HPPK DS digital signature scheme. Using their linear configuration, MPPK KEM, and HPPK DS produce compact public keys (196-536 bytes) and signatures (144-272 bytes) – considerably smaller than ML-DSA's 2–3 KB. We present a blockchain design that natively integrates these primitives. We analyze its security under classical and quantum adversarial models. We estimate transaction throughput and block capacity. Compared

Quantum Information Technologies Journal, Vol. 2_1, 117–138.

doi: 10.13052/qitj2795-0492.217

© 2026 River Publishers

to lattice-based alternatives, our projections suggest that compact representations significantly improve transaction throughput and reduce on-chain storage. This work aims to contribute to ongoing quantum-safe standardization efforts, including those within ETSI, by demonstrating a concrete architectural pathway.

Keywords: Blockchain, quantum-safe cryptography, multivariate polynomial public key, homomorphic polynomial public key, key encapsulation mechanism, digital signature.

1 Introduction

Blockchains rely on public-key cryptography for user identities, transaction signatures, and secure peer-to-peer communications. In current systems, the dominant signature schemes are Elliptic Curve Digital Signature Algorithm (ECDSA) (used in Bitcoin and Ethereum) and Ed25519¹ (widely used in modern protocols such as Transport Layer Security (TLS) and distributed systems), while Elliptic Curve Diffie-Hellman (ECDH) is the standard mechanism for key agreement and secure channel establishment [1, 7, 9]. The security of these schemes is based on the hardness of the elliptic curve discrete logarithm problem, which is not efficient to compute classically, but becomes vulnerable under quantum computers via Shor’s algorithm [14].

Although large-scale fault-tolerant quantum computers are not yet available, the so-called “harvest now, decrypt later” threat model has already motivated migration toward Post-Quantum Cryptography (PQC). In this model, adversaries record encrypted blockchain traffic today and decrypt it once the quantum capability becomes available.

National Institute of Standards and Technology (NIST) has completed multiple rounds of PQC evaluation. It has standardized ML-KEM (Module-Lattice-based Key Encapsulation Mechanism, formerly CRYSTALS-Kyber) and ML-DSA (Module-Lattice-based Digital Signature Algorithm, formerly CRYSTALS-Dilithium) as the primary lattice-based schemes, while FALCON and SPHINCS+ remain alternate finalists [10–13]. However, these schemes introduce significant overhead compared to classical elliptic-curve systems.

¹Ed stands for Edwards-curve Digital Signature Algorithm (EdDSA), 255 means that the algorithm utilizes a 255-bit (32-byte) private key, and 19 is the prime number $2^{255} - 19$ that sets the mathematical boundary for the curve.

ML-DSA signatures range from approximately 2.4 to 3.3 KB, while ML-KEM public keys and ciphertexts are on the order of 800 to 1,568 bytes depending on the security level [13]. Importantly, in blockchain systems, this overhead is amplified because: (i) every signature is permanently stored on-chain, (ii) every transaction is replicated across thousands of nodes, and (iii) many systems additionally rely on public-key encryption or key encapsulation for secure off-chain communication or confidential transaction layers.

Therefore, not only signature size but also Key Encapsulation Mechanism (KEM) ciphertext and public-key size directly affect scalability, storage cost, and network propagation latency.

Multivariate polynomial cryptography is an alternative family of PQC. Its security is based on the hardness of solving systems of nonlinear multivariate equations over finite fields, which is related to the NP-hard modular Diophantine equation problem [8]. In this line of work, Kuang et al. proposed the schemes Multivariate Polynomial Public Key (MPPK) KEM and Homomorphic Polynomial Public Key (HPPK) Digital Signature (DS) [4–6]. They operate on a large prime field $\text{GF}(p)$ and a hidden modular ring $\mathbb{Z}/t\mathbb{Z}$, enabling compact key and signature representations. Moreover, their linear instantiation eliminates the need to find roots over extension fields, which leads to faster encapsulation and verification while preserving the security properties. We refer the reader to the original publications for the underlying security arguments and a full cryptanalysis of MPPK KEM and HPPK DS [4–6]. In the following, we focus on the system-level integration of these two schemes into a blockchain architecture, their blockchain-specific security, and performance implications.

In this article, we propose a new quantum-safe blockchain architecture that integrates MPPK KEM and HPPK DS. In contrast to retrofit approaches that replace only signatures or encryption layers, our design is purpose-built around compact multivariate primitives. We demonstrate how transactions are signed using HPPK DS, how optional confidentiality is achieved using MPPK KEM, and how block validation remains efficient at scale. We further analyze security against quantum and classical adversaries, including forgery resistance, malleability, and key-substitution attacks.

The remainder of this article is organized as follows. Section 2 reviews the fundamentals of the blockchain and motivates PQC constraints. Section 3 introduces the MPPK KEM and HPPK DS schemes. Section 4 describes the design of the blockchain. Security and performance analyses are given in Sections 5 and 6, respectively. Section 6.4 discusses the relevance with respect

to European Telecommunications Standards Institute (ETSI) standardization efforts. Section 7 concludes the article.

2 Background, Motivation, and Related Work

2.1 Blockchain Operations

A blockchain is a distributed ledger of transactions grouped into blocks. Users control accounts (or unspent transaction outputs) via public-private key pairs. Before inclusion in a block, a transaction is authorized by a DS created with the sender's private key and verified by the network nodes using the corresponding public key.

In addition to DSs, some blockchain systems also employ KEMs to enable encrypted communication channels or confidential transactions. Consequently, two cryptographic primitives are fundamental:

- **DS:** ensures authenticity, integrity, and non-repudiation of transactions.
- **KEM:** enables secure establishment of symmetric keys for confidential communications.

Hash functions, e.g., SHA-256 and SHA-3, are generally considered quantum-safe against Grover's algorithm [3] when appropriately parameterized and are not the primary scalability bottleneck.

2.2 Blockchain Architecture Overview

In a typical blockchain system, transactions are sent by broadcast over a peer-to-peer network and grouped into blocks by designated nodes, miners or validators. Each block contains a set of transactions, a cryptographic hash of the previous block, and a commitment, e.g., a Merkle root, over all included transactions.

Consensus mechanisms, such as Proof-of-Work (PoW) or Proof-of-Stake (PoS), are used to ensure agreement on the next valid block and maintain a consistent global ledger state across distributed nodes [9, 15].

All validating nodes must verify DSs before accepting transactions. As a result, both computational cost (signature verification) and data size (signatures, public keys, and optional ciphertexts) directly influence throughput, latency, and long-term storage requirements. These constraints become increasingly critical when transitioning to post-quantum cryptographic primitives, which typically introduce larger key and signature sizes.

2.3 Why NIST Lattice Candidates Are Not Ideal for Blockchains

Table 1 compares representative PQC schemes at NIST security Level V (approximately equivalent to AES-256).

Data are taken from standard specifications and from Ref. [5]. ML-DSA signatures exceed three KB, implying that a one MB block can contain only a few hundred transactions. In contrast, HPPK DS signatures of 272 bytes allow approximately 4,000 transactions per block, yielding an order-of-magnitude improvement in throughput before considering network overhead. Similarly, the public keys and ML-KEM ciphertexts are significantly larger than those of MPPK KEM. In large-scale blockchain systems with millions of accounts, this difference directly translates into increased state size, storage cost, and synchronization overhead.

We do not claim that lattice-based schemes are unsuitable for all blockchains; rather, for systems where transaction throughput and storage are primary constraints, the compactness of multivariate primitives offers a compelling alternative.

Another important distinction lies in how the security levels are parameterized. Lattice-based schemes such as ML-KEM and ML-DSA define discrete parameter sets, e.g., ML-KEM-512/768/1024 and ML-DSA-2/3/5, each corresponding to a fixed security level with fixed performance characteristics [10–12]. Transitioning between security levels requires switching to a different set of parameters, resulting in incremental increases in bandwidth and storage requirements.

In contrast, multivariate polynomial schemes such as MPPK KEM and HPPK DS enable security scaling through algebraic parameters such as field size (p) and the number of variables (m). This provides a continuous trade-off between security and efficiency without altering the underlying algorithmic structure. Such flexibility is particularly advantageous in blockchain environments, where even small increases in cryptographic overhead can significantly impact global system scalability.

Table 1 Comparison of public key and signature/ciphertext sizes (bytes) at NIST security Level V

Scheme	Public key	Signature/Ciphertext
ML-DSA (Level V)	1,600	3,293
ML-KEM (Level V)	1,568	1,568 (ciphertext)
HPPK DS (Level V)	356	272
MPPK KEM (Level V)	350	350 (ciphertext)

2.4 Threat Model

Classical public-key cryptography relies on hardness assumptions such as integer factorization and the discrete logarithm problem of the elliptic curve, which underpin widely deployed schemes such as Rivest-Shamir-Adleman (RSA), ECDSA, and ECDH. These assumptions are believed to be intractable for classical adversaries but can be efficiently broken by sufficiently powerful quantum computers using Shor’s algorithm [14].

In this work, we adopt a PQC threat model in which adversaries are assumed to possess both classical and quantum computational capabilities. We consider the “harvest now, decrypt later” scenario, where encrypted data recorded today may be stored and later decrypted once large-scale quantum computers become available. DSs are not vulnerable to retrospective decryption; they provide authenticity that remains valid even after quantum computers arrive. However, the confidentiality of any encrypted payload, e.g., via MPPK KEM, is the primary concern for harvest-now attacks.

This distinction is especially relevant in blockchain systems, where historical data are permanently recorded, publicly accessible, and cannot be retroactively replaced or re-encrypted. Our objective is therefore to ensure long-term confidentiality and authenticity over decades without relying on optimistic assumptions about the timeline of quantum computing development.

2.5 Related Work

Various PQC blockchain designs have been proposed. The Quantum Resistant Ledger (QRL) employs eXtended Merkle Signature Scheme (XMSS), a stateful hash-based signature scheme. Although secure under well-established assumptions, XMSS introduces significant operational complexity due to state management requirements and produces signatures that typically exceed two KB, which limits scalability in highthroughput environments.

Lattice-based approaches based on ML-KEM and ML-DSA are currently the NIST reference standards for post-quantum key encapsulation and DSs. However, their relatively large key and signature sizes introduce non-negligible storage and bandwidth overhead in blockchain systems, particularly when every transaction must be permanently replicated across distributed nodes.

Multivariate signature schemes such as Rainbow have also been studied in the literature. However, several variants have been broken through

cryptanalytic attacks, including those reported in Ref. [2]. These results highlight that not all multivariate constructions provide robust long-term security.

In contrast, the MPPK/HPPK family is based on hidden-ring multivariate constructions combined with structured modular transformations. To date, no practical cryptanalytic break has been reported against these schemes under their intended parameter regimes.

To our knowledge, this work is the first to propose a blockchain architecture explicitly built around MPPK KEM and HPPK DS, leveraging their compact representations to achieve scalable post-quantum ledger design.

3 Multivariate Polynomial Primitives: MPPK KEM and HPPK DS

This section introduces the two core cryptographic primitives used in the proposed blockchain architecture: the MPPK KEM and HPPK DS schemes. Both constructions operate over a large finite field $\text{GF}(p)$ and a hidden modular ring $\mathbb{Z}/t\mathbb{Z}$. Both derive their security from the hardness of solving structured systems of multivariate polynomial equations over finite fields.

We adopt throughout this work the linear private polynomials configuration. It provides the most efficient instantiation without weakening the underlying security reduction. We refer the reader to Refs. [4–6] for detailed security arguments and a complete cryptanalysis.

3.1 MPPK KEM – Linear Case

MPPK KEM [4, 6] is a public-key encryption-like functionality that enables two parties to derive a shared secret over an insecure channel. The scheme is defined over a finite field $\text{GF}(p)$ and a hidden modulus ring $\mathbb{Z}/t\mathbb{Z}$, where t is the hidden modulus. We present the scheme through the KEM interface, that is, the functions KeyGen, Encap, and Decap.

3.1.1 Key generation (KeyGen)

KeyGen, the key generation function, consists of the following elements.

1. Sample a structured base polynomial

$$\beta(x_0, x_1, \dots, x_m) = \sum_{i=0}^n \sum_{j=1}^m c_{ij} x_0^i x_j, \quad c_{ij} \in \text{GF}(p).$$

2. Define two linear private polynomials

$$f(x_0) = f_0 + f_1x_0, \quad h(x_0) = h_0 + h_1x_0. \quad (1)$$

3. Compute the composed polynomials

$$\varphi(x) = \beta(x_0, x_1, \dots, x_m) \cdot f(x), \quad \psi(x) = \beta(x_0, x_1, \dots, x_m) \cdot h(x).$$

4. Publish their truncated forms Φ and Ψ by removing the selected constants and highest-degree terms in x_0 .
5. Select random values $R_0, R_n \in \mathbb{Z}/t\mathbb{Z}$ such that $\gcd(R_0, t) = \gcd(R_n, t) = 1$, and define masking terms

$$N_0 = R_0\beta_0 \bmod t, \quad N_n = R_n\beta_n x_0^{n+1} \bmod t.$$

The public key is:

$$\text{pk} = (\Phi, \Psi, N_0, N_n),$$

and the private key is:

$$\text{sk} = (f_0, f_1, h_0, h_1, R_0, R_n, t).$$

3.1.2 Encapsulation (encap)

Encap encapsulates a shared secret $s \in \text{GF}(p)$ as follows:

1. Sample randomly the values $r_1, \dots, r_m \in \text{GF}(p)$.
2. Evaluate the public truncated polynomials Φ and Ψ at the point $(s, r_1, \dots, r_m)$, obtaining values $\bar{\Phi}$ and $\bar{\Psi}$.
3. Compute masked components

$$\bar{N}_0 = N_0(s), \quad \bar{N}_n = N_n(s),$$

where N_0 and N_n are evaluated at the same secret s .

4. Output the ciphertext

$$C = (\bar{\Phi}, \bar{\Psi}, \bar{N}_0, \bar{N}_n).$$

The shared secret is derived deterministically from s , e.g., via a key derivation function.

3.1.3 Decapsulation (decap)

Decap is given ciphertext $C = (\bar{\Phi}, \bar{\Psi}, \bar{N}_0, \bar{N}_n)$ and performs the following decapsulation operations.

1. Recover $\bar{\beta}_0 = \bar{N}_0 \cdot R_0^{-1} \bmod t$ and $\bar{\beta}_n = \bar{N}_n \cdot R_n^{-1} \bmod t$.

2. Reconstruct intermediate polynomials φ and ψ using the recovered β components and the public Φ, Ψ .
3. Compute

$$k = \frac{\varphi}{\psi} \quad (\text{in the field } \text{GF}(p)).$$

4. Solve the linear relation for s

$$s = \frac{kh_0 - f_0}{f_1 - kh_1} \quad (\text{in } \text{GF}(p)).$$

3.1.4 Security

According to the analysis in Refs. [4, 6], security is based on the hardness of solving modular Diophantine equations over finite fields [8]. In the linear setting, the best-known attack has time complexity is $O(p^{m-3})$, which yields 128-256-bit security depending on the choice of (p, m) . The scheme achieves Indistinguishability under Chosen-Plaintext Attack (IND-CPA) security. Indistinguishability under Adaptive Chosen-Ciphertext Attack (IND-CCA2) security can be obtained via standard transformations if required, e.g., Fujisaki-Okamoto.

3.2 HPPK Digital Signature

HPPK DS [5] extends the MPPK framework with authentication and non-repudiation support. To conceal the underlying ring parameters, we introduce a structured verification equation based on modular masking and Barrett reduction. We present the scheme using the interface functions KeyGen, Sign, and Verify.

3.2.1 Key Generation (KeyGen)

Key generation consists of the following steps.

1. As in Equation (1), select linear private polynomials

$$f(x), h(x) \in \text{GF}(p)[x].$$

2. Choose secret scalars R_1, R_2 and moduli S_1, S_2 .
3. Compute masked coefficients

$$P_{ij} = R_1 p_{ij} \bmod S_1, \quad Q_{ij} = R_2 q_{ij} \bmod S_2.$$

4. Apply the Barrett transformation to derive public parameters

$$(p'_{ij}, q'_{ij}, s_1, s_2, \mu_{ij}, v_{ij}),$$

where s_1, s_2 are Barrett reduction constants and μ_{ij}, v_{ij} are precomputed scaling factors.

The public key is

$$\text{pk} = (p'_{ij}, q'_{ij}, s_1, s_2, \mu_{ij}, v_{ij}),$$

and the private key is

$$\text{sk} = (f, h, R_1, R_2, S_1, S_2).$$

3.2.2 Signing (sign)

Given a message digest $x \in \text{GF}(p)$, a DS is calculated according to the following steps.

1. Sample a fresh random value $a \in \text{GF}(p)$ per signature.
2. Compute signature components

$$F = R_2^{-1}(af(x) \bmod p) \bmod S_2, \quad H = R_1^{-1}(ah(x) \bmod p) \bmod S_1.$$

The signature is $\sigma = (F, H)$.

3.2.3 Verification (verify)

Given message digest x and signature $\sigma = (F, H)$, the verifier accepts when the following equality holds for all $j = 1, \dots, m$:

$$\sum_i \left(Fq'_{ij} - s_2 \left\lfloor \frac{Fv_{ij}}{R} \right\rfloor \right) x^i = \sum_i \left(Hp'_{ij} - s_1 \left\lfloor \frac{H\mu_{ij}}{R} \right\rfloor \right) x^i \pmod{p},$$

where $\lfloor \cdot \rfloor$ denotes integer division with floor, and R is a public constant derived from the Barrett reduction parameters (the same R used during key generation). For full detail on the Barrett transformation and the exact definitions of $s_1, s_2, \mu_{ij}, v_{ij}, R$, we refer the reader to Ref. [5].

3.2.4 Security

As established in Ref. [5], HPPK DS achieves Existential Unforgeability under Chosen-Message Attack (EUF-CMA) security, stemming from the hardness of recovering hidden ring parameters from masked polynomial structures. The best-known attack time complexity is $O(2^L)$, for key recovery, and $O(2^{2L})$, for forgery. For $L \in \{144, 208, 272\}$, the scheme targets the respective NIST security Levels I, III, and V. Currently, no polynomial-time quantum attacks are known on the underlying multivariate problem.

We emphasize that while MPPK KEM and HPPK DS offer compelling performance characteristics for blockchain applications, their security has not yet been validated through the same extensive public cryptanalysis as NIST-standardized ML-KEM and ML-DSA. NIST’s multi-year evaluation process involved extensive cryptanalysis by the global cryptographic community, whereas the MPPK/HPPK family represents a more recent proposal that requires continued security evaluation. The security arguments presented in this work rely on the hardness of solving structured multivariate polynomial systems over finite fields, a problem believed to be computationally hard. However, as with any cryptographic primitive, the long-term security of these schemes depends on continued cryptanalytic scrutiny.

3.2.5 Why choose linear private polynomials?

Experimental analysis, presented in Ref. [6], shows that the linear private polynomial configuration, Equation (1), provides a significant performance advantage over higher-degree variants, particularly in decapsulation and verification, where root-finding in extension fields is avoided. According to the authors, the security reduction is independent of the private polynomials degree, which means that performance can be improved without weakening the underlying hardness assumptions. For this reason, the linear configuration is adopted throughout this work.

3.3 Security Summary for Blockchain Deployment

We summarize the security properties of the two cryptographic schemes in the context of blockchain deployment.

3.3.1 MPPK KEM

MPPK KEM provides IND-CPA security under the hardness of structured multivariate equation solving over finite fields [8]. It ensures confidentiality of optional encrypted payloads such as transaction metadata or off-chain commitments. In blockchain environments, where ciphertexts are not repeatedly queried through decryption oracles, IND-CPA security is sufficient in practice.

3.3.2 HPPK DS

HPPK DS reaches EUF-CMA security. This ensures that generation of valid signatures requires knowledge of the private key. The design eliminates classical malleability vectors and resists algebraic reconstruction attacks due

to the nonlinear masking introduced by the Barrett reduction and hidden modulus structures [5].

3.3.3 Quantum computing resistance

Neither scheme is known to be vulnerable to efficient quantum attacks. Unlike integer factorization or discrete logarithms, the underlying multivariate polynomial problems currently do not have polynomial-time quantum algorithms. However, as with any non-standardized scheme, continued cryptanalysis is necessary.

3.3.4 Blockchain suitability

In blockchain systems, signatures are permanently stored and publicly verified. Therefore, EUF-CMA security is essential for authentication, while compactness and verification efficiency directly affect scalability. The combination of HPPK DS and MPPK KEM provides a balanced foundation for the design of post-quantum blockchains, fulfilling both efficiency and long-term security.

4 A New Quantum-Safe Blockchain Architecture

We propose a blockchain architecture designed building on two compact post-quantum schemes: MPPK KEM and HPPK DS. They are the sole public-key cryptographic building blocks of the system.

The consensus layer, e.g., PoW or PoS, is intentionally decoupled from the cryptographic layer and may follow any standard protocol without modification. Our contribution focuses on the cryptographic and transaction validation substrate.

4.1 Account Model

Each user controls a wallet consisting of two key pairs.

- An HPPK DS key pair for authentication and transaction signing.
- An optional MPPK KEM key pair for confidentiality and secure message encapsulation.

Let HPPK PK and MPPK PK denote the corresponding HPPK DS and MPPK KEM public keys. To bind both cryptographic identities to a single blockchain account, the address is defined as:

$$\text{Address} = \text{SHA256}(\text{HPPK-PK} \parallel \text{MPPK-PK}), \quad (2)$$

truncated to the required length. Although SHA-256 offers 128-bit postquantum collision resistance against Grover's algorithm, i.e., 2^{128} operations. If a stronger hash function is required, the protocol can be upgraded, e.g., to SHA-3 with larger output. This construction ensures that both the authentication and the encryption identities are cryptographically coupled to the same on-chain identifier, simplifying wallet management while preserving the separation of cryptographic roles.

4.2 Transaction Format

A transaction T consists of the following fields.

1. **Metadata:** protocol version and transaction fee.
2. **Inputs:** references to previous unspent transaction outputs (UTXOs) or account-based state entries, nonce.
3. **Outputs:** recipient addresses and transfer amounts.
4. **Optional encrypted payload:** ciphertext produced using MPPK KEM encapsulating a symmetric key for encrypting sensitive transaction data, e.g., memo, amount, or contract-specific data.
5. **Signature:** an HPPK DS over all preceding fields (serialized canonically).

The signature is computed using the sender's private key and a fresh random value a for each transaction. While HPPK DS incorporates randomness to ensure security (nonce reuse resistance), the signature structure prevents algebraic manipulation of signature components without detection during verification. This protects against malleability attacks that alter valid signatures to produce different but still valid encodings, since the verification equation is structurally non-linear and coupling constraints prevent component substitution.

4.3 Block Structure

Each block contains two items.

- **Header:** previous block hash, Merkle root of transactions, timestamp, consensus-related data, e.g., nonce in PoW or validator signature in PoS, and a state commitment.
- **Body:** ordered list of validated transactions.

Additional cryptographic primitives beyond HPPK DS and MPPK KEM are not introduced at the protocol level, ensuring a minimal and uniform cryptographic footprint.

4.4 Verification and Block Validation

Upon receiving a transaction, a node performs the following steps.

1. **State validation:** verify that all referenced inputs exist and are unspent.
2. **Signature verification:** validate the HPPK DS signature using the sender's public key stored in the global state. According to the Ref. [5], verification requires only a small number of modular multiplications and Barrett reductions, with an estimated cost of approximately 22k CPU cycles.
3. **Optional decryption handling:** if an encrypted payload is present, only authorized recipients possessing the corresponding MPPK KEM private key may decapsulate and decrypt; all other nodes treat the ciphertext as opaque data.

A block is considered valid when all included transactions pass verification and the consensus protocol is satisfied. The verification process is independent of the consensus mechanism; only the validity of signatures and state transitions is checked. Due to the lightweight nature of signature verification, system performance is primarily constrained by network propagation and storage rather than cryptographic computation.

4.5 Confidential Transactions Using MPPK KEM

Transaction confidentiality is optionally supported through MPPK KEM as follows.

1. The sender obtains or generates a MPPK KEM public key for the recipient.
2. A random symmetric key K is encapsulated using MPPK KEM and the public key, producing ciphertext C .
3. The sensitive payload is encrypted using K via a symmetric authenticated encryption scheme, e.g., Advanced Encryption Standard in Galois/Counter Mode (AES-GCM),
4. Both C and the encrypted payload are included in the transaction.

Only the intended recipient can recover K , decapsulating and decrypting the payload. This design enables optional confidentiality without introducing zero-knowledge proofs or additional interactive protocols.

4.5.1 Validation of confidential data

An important consideration is how blockchain nodes validate transactions containing encrypted payloads. We distinguish between *validation-critical*

data and *confidential data*. For the validation-critical data, transaction inputs, output addresses, amounts, and fees are always included in the clear (unencrypted portion) of the transaction. These fields are essential for consensus and state validation. The HPPK DS signature is computed over these fields, ensuring authenticity and integrity. Regarding confidential data, additional metadata, memos, contract parameters, or custom data may be encrypted using MPPK KEM. This data is *not* required for consensus or validation. Nodes validate the transaction based on the cleartext validation-critical data, treating the encrypted payload as an opaque blob.

The validation procedure for confidential transactions is as follows. (i) Nodes verify the HPPK DS signature over the clear-text portion (which includes transaction inputs, outputs, and amounts but excludes the encrypted payload). (ii) Nodes validate that inputs are unspent and outputs are valid. (iii) The encrypted payload is checked for syntactical correctness (ciphertext format) but is not decrypted. (iv) Only the intended recipient (or authorized parties with the MPPK KEM private key) can decrypt the payload.

This design ensures the following properties. Consensus nodes can validate transactions without accessing encrypted data. The encrypted payload does not affect state transitions. Confidentiality is preserved while maintaining blockchain integrity. A transaction signature binds both the clear-text and encrypted portions to prevent substitution attacks.

5 Security Analysis in the Blockchain Context

5.1 Quantum Security Model

We adopt a post-quantum adversarial model in which attackers possess both classical and quantum computational capabilities. We consider the “harvest-now, decrypt-later” scenario, where all on-chain encrypted data, i.e., MPPK KEM ciphertexts, are assumed to be permanently recorded and potentially decrypted in the future, once scalable quantum computers become available. DSs are not vulnerable to retrospective decryption. They provide authenticity that remains valid even after quantum computers arrive. Nevertheless, the size of the signature and cost of verification directly affect the scalability of a blockchain.

Both MPPK KEM and HPPK DS rely on the hardness of solving structured modular polynomial systems, for which no efficient quantum algorithms are known to date. According to the analyzes in Refs. [4, 5],

the best-known attack remains exponential time complexity, in the relevant security parameters.

For MPPK KEM, security scales with parameters (p, m) , with time complexity approximately $O(p^{m-3})$. For HPPK DS, security scales with the hidden ring size L , resulting in attack complexities of $O(2^L)$ for key recovery and $O(2^{2L})$ for forgery. Parameter selections such as $p = 2^{64}$, and $L = 144$, are projected to provide at least 128-bit security, exceeding the NIST Level I requirements.

5.2 Forgery Resistance and Malleability

As established in Ref [5], the HPPK DS scheme achieves EUF-CMA. Forgery requires simultaneously solving coupled nonlinear constraints induced by the Barrett reduction structure, making algebraic manipulation ineffective.

In contrast to classical schemes such as ECDSA, where signature components can sometimes be algebraically manipulated, HPPK DS signatures are structurally non-malleable. The signature is uniquely bound to both the message and private key. Internal randomness is not exposed in signatures.

5.3 Key Substitution Resistance

Key substitution attacks aim to find alternative public keys that validate a given signature. In this architecture, such attacks are mitigated at two levels. Firstly, the blockchain address is derived from a cryptographic hash of the public keys, see Equation (2), making substitution hard without hash collision 128-bit post-quantum security against Grover's algorithm. Secondly, the HPPK DS verification depends explicitly on structured public coefficients, preventing algebraic equivalence between distinct key pairs. Under these two assumptions on hash collision resistance, key substitution is computationally infeasible.

5.4 Long-Term Key Security and Rotation

As in classical blockchain systems, compromised private key results in loss of control over associated funds. To mitigate long-term risks, periodic key rotation is recommended, in which users migrate assets to fresh key pairs. A notable advantage of HPPK DS is its compact key structure, which makes frequent key rotation practical with minimal storage and bandwidth overhead, supporting long-lived blockchain systems under evolving cryptographic assumptions.

6 Performance Projections

6.1 Benchmark Data from Literature

Table 2 summarizes the median CPU cycles for MPPK KEM (linear configuration) and HPPK DS, extracted from Refs. [5, 6]. The results are provided for NIST security Levels I, III, and V.² All measurements were obtained using the SUPERCOP benchmarking framework on an Intel i7-10700 CPU, providing a consistent basis for cross-scheme comparison. Actual performance in a production blockchain environment may vary due to network latency, disk I/O, and software implementation details. The results indicate that both primitives operate within tens of thousands of CPU cycles even at the highest security level, suggesting they are suitable for high-throughput distributed systems such as blockchains, subject to further implementation optimization.

6.2 Transaction Throughput Estimation

Consider a representative validator node equipped with a single-core CPU operating at three GHz. At security Level V, HPPK DS verification requires approximately $2.2 \cdot 10^4$ cycles per signature [5]. The single-core performance is:

$$\frac{3.0 \cdot 10^9}{2.2 \cdot 10^4} \approx 1.36 \cdot 10^5$$

signature verifications per second on a single core.

Now, consider a representative validator node equipped with a four-core CPU operating at three GHz. The four-core performance is (ideal scaling):

$$1.36 \cdot 10^5 \times 4 \approx 5.45 \cdot 10^5$$

signature verifications per second.

Table 2 Median CPU cycles for MPPK KEM and HPPK-DS [5, 6]

Scheme	Level I	Level III	Level V
MPPK KEM key generation	20,409	25,696	42,355
MPPK KEM encapsulation	36,337	12,510	16,046
MPPK KEM decapsulation	34,771	18,349	22,285
HPPK DS signing	12,510	14,382	16,046
HPPK DS verification	18,349	21,145	22,285

²Level I (III or V) is at least as hard to break as a 128 (192 or 256)-bit key search on a block cipher such as AES-128 (192 or 256).

In practice, performance is limited by memory bandwidth, cache misses, branch mispredictions, operating system overhead, and network I/O. With conservative assumptions (50–60% efficiency), we estimate practical throughput of approximately $2 - 3 \times 10^5$ verifications per second on a four-core system. At this rate, the cryptographic verification cost is unlikely to be the primary bottleneck in most blockchain deployments, where network propagation and state storage dominate.

6.3 Block Capacity

At security Level V, each HPPK DS signature occupies approximately 272 bytes.

However, realistic block capacity must account for transaction overhead beyond signatures. A typical transaction includes

- a signature of 272 bytes,
- transaction inputs (two inputs are typical) of approximately 160 bytes,
- transaction outputs (two outputs are typical) of approximately 200 bytes, and
- metadata (version, fee, transaction nonce) of approximately 80 bytes,

for a total per transaction of approximately 712 bytes for a block size of two MB, we have

$$\frac{2 \cdot 10^6}{712} \approx 2,800 \text{ transactions per block (theoretical).}$$

A realistic estimate, accounting for variable input/output counts and block header overhead, is approximately 2,000 to 2,500 transactions per block. The theoretical upper bound ignoring overhead (for reference only) is

$$\frac{2 \cdot 10^6}{272} \approx 7,350 \text{ transactions.}$$

This performance is comparable to classical blockchain systems. For example, one MB Bitcoin block typically contains of the order of a few thousand ECDSA signatures (64 bytes each) plus transaction overhead. The proposed architecture thus maintains competitive block utilization while simultaneously providing post-quantum security and optional confidentiality.

6.4 Migration Path Toward ETSI Standardization

ETSI has initiated efforts to standardize PQC migration strategies, including those for distributed ledgers. The architecture presented here is intended to illustrate one possible design that could serve as an input for ongoing ETSI discussions on quantum-safe blockchains.

Migrating existing systems to this architecture would require a protocol-level transition, e.g., a hard fork or a layered migration. As the proposed design is a standalone blockchain, backward compatibility with classical chains is not assumed. To facilitate interoperability, a bridge mechanism could be defined in which classical assets are locked in legacy chains and equivalent assets are minted in the quantum-safe chain, with state transitions authenticated by HPPK DS signatures. Detailed specification of such a bridge is left for future work.

7 Conclusion

We have presented a quantum-safe blockchain architecture that replaces classical primitives of elliptic-curves, such as ECDSA, Ed25519, and ECDH, with compact multivariate constructions, namely MPPK KEM and HPPK DS. In the linear private polynomials instantiation, the proposed schemes achieve small public keys (196-536 bytes) and compact signatures (144-272 bytes), while their security is grounded in the hardness of solving modular multivariate polynomial systems.

Benchmark results from the literature indicate that both key encapsulation and signature operations execute within tens of thousands of CPU cycles, which would enable high-throughput blockchain operation if the primitives are implemented efficiently. The architecture also supports optional confidentiality through encryption based on MPPK KEM, avoiding the need for zero-knowledge proofs or heavy protocol extensions.

We acknowledge that MPPK KEM and HPPK DS have not yet undergone the same level of public cryptanalysis as NIST-standardized ML-KEM and ML-DSA. NIST's multi-year evaluation process involved extensive cryptanalysis by the global cryptographic community, whereas the MPPK/HPPK family is a more recent proposal that still requires ongoing security evaluation. However, their compactness and performance make them attractive candidates for further study in blockchain systems. We hope that this work

motivates additional analysis and standardization efforts. We intend to contribute to the design as input to ongoing ETSI activities in quantum-safe distributed ledgers.

References

- [1] Daniel J. Bernstein, Niels Duif, Tanja Lange, Peter Schwabe, and Bo-Yin Yang. Highspeed high-security signatures. In *Cryptographic Hardware and Embedded Systems – CHES 2011*, volume 6917 of *Lecture Notes in Computer Science*, pages 124–142. Springer, 2012.
- [2] Jintai Ding, John Deaton, Daniel S. Schmidt, V. Vishakha, and Zhenfei Zhang. Cryptanalysis of the lifted unbalanced oil vinegar signature scheme. In *Advances in Cryptology – CRYPTO 2020*, volume 12171 of *Lecture Notes in Computer Science*, pages 279–298. Springer, 2020.
- [3] Lov K Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219, 1996.
- [4] Randy Kuang, Maria Perepechaenko, and Michel Barbeau. A new post-quantum multivariate polynomial public key encapsulation algorithm. *Quantum Information Processing*, 21(10):360, 2022.
- [5] Randy Kuang, Maria Perepechaenko, Mahmoud Sayed, and Dafu Lou. Homomorphic polynomial public key with Barrett transformation for digital signature. *Academia Quantum*, 1(1), 2024.
- [6] Randy Kuang, Maria Perepechaenko, Ryan Toth, and Michel Barbeau. Benchmark performance of the multivariate polynomial public key encapsulation mechanism. In Slim Kallel, Mohamed Jmaiel, Mohammad Zulkernine, Ahmed Hadj Kacem, Frederic Cuppens, and Nora Cuppens, editors, *Risks and Security of Internet and Systems*, pages 239–255, Cham, 2023. Springer Nature Switzerland.
- [7] Adam Langley, Mike Hamburg, and Sean Turner. Elliptic curves for security (curve25519 and curve448). RFC 7748, 2016. IETF Standard.
- [8] Christopher Moore and Stephan Mertens. *The Nature of Computation*. Oxford University Press, 2011.
- [9] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. <http://bitcoin.org/bitcoin.pdf>, 2008. Accessed: 2026-04-30.
- [10] National Institute of Standards and Technology. Status report on the third round of the NIST post-quantum cryptography standardization process. NIST Interagency Report NIST IR 8413, National Institute of Standards and Technology, 2022.

- [11] National Institute of Standards and Technology, Gorjan Alagic, Quynh Dang, Dustin Moody, Angela Robinson, Hamilton Silberg, and Daniel Smith-Tone. Module-latticebased key-encapsulation mechanism standard. Technical Report NIST FIPS 203, National Institute of Standards and Technology, August 2024. Published August 13, 2024.
- [12] National Institute of Standards and Technology, Thinh Dang, Jacob Lichtinger, Yi-Kai Liu, Carl Miller, Dustin Moody, Rene Peralta, Ray Perlner, and Angela Robinson. Module-lattice-based digital signature standard. Technical Report NIST FIPS 204, National Institute of Standards and Technology, August 2024. Published August 13, 2024.
- [13] National Institute of Standards and Technology. Post-quantum cryptography standardization: Selected algorithms. <https://csrc.nist.gov/projects/post-quantum-cryptography> 2024. Accessed: 2026-04-30.
- [14] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Review*, 41(2):303–332, 1999.
- [15] Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger. <https://ethereum.github.io/yellowpaper/paper.pdf>, 2014.

Biographies



Michel Barbeau is a professor of Computer Science at Carleton University. He received his Ph.D. in Computer Science from Université de Montreal, Canada in 1991. From 1991 to 1999, he was a professor at Université de Sherbrooke. During the '98–'99 academic year, he was a visiting researcher at the University of Aizu, Japan. Since 2000, he has worked at Carleton University, School of Computer Science, Canada. Michel Barbeau's primary area of expertise is computer networks, specifically architecture and protocols. Research interests include quantum computing, underwater communications and networks, drones, quantum algorithms, and network control systems. Michel Barbeau has numerous publications in the field of AI and quantum computing. Michel Barbeau is a member of the NATO

Science and Technology Organization (STO) Exploratory Team TSI ET 008: Enabling NATO Resilience Through Quantum Secure Technologies and a member of the QuARC (Quantum Advancement Research Centre) at Carleton University.



Randy Kuang is Co-Founder and Chief Scientist of Quantropi Inc., where he leads research and development in post-quantum cryptography and quantum-safe communications. He holds a Ph.D. in Atomic and Molecular Physics from Memorial University of Newfoundland. Prior to founding Quantropi, Dr. Kuang held a senior research position at Nortel Networks, where he worked on next-generation networking and security systems, and later served as Co-Founder and Chief Technology Officer at inBay Technologies, leading the design and deployment of a successful cybersecurity platform. He is the inventor or co-inventor of more than 40 U.S. patents, with foundational contributions including the Quantum Permutation Pad (QPP) for quantum symmetric encryption, the Homomorphic/Multivariate Polynomial Public Key (HPPK/MPPK) framework for postquantum key encapsulation and digital signatures, and the Quantum Encryption in Phase Space (QEPS) scheme for coherent optical communications. His work on algebraic cryptography and search-based security has been published in leading journals, and he serves on the editorial boards of *EPJ Quantum Technology*, *Scientific Reports* (Nature Portfolio), and *Academia Quantum*. He is widely recognised for bridging rigorous theoretical foundations with deployable cryptographic systems.