

Quantum-adaptive lattice signatures for next-generation secure communications

Maksim Iavich^{1,*}

¹*Caucasus University, Paata Saakadze Str., 1, Tbilisi, 0102, Georgia*

Abstract

The maturation of quantum computing threatens the asymmetric cryptographic primitives – RSA and elliptic-curve signatures in particular – that currently secure the overwhelming majority of network communications. NIST has standardized lattice-based (CRYSTALS-Dilithium, FALCON) and hash-based (SPHINCS+) post-quantum digital signature schemes, but deployed systems must still choose a single, fixed operating point on the trade-off between signature size, key size, signing/verification latency, and long-term security margin, even though the resource constraints and threat exposure of a TLS server, a blockchain validator, and a battery-powered IoT sensor differ by orders of magnitude. This paper proposes a quantum-adaptive lattice signature framework that dynamically selects Verkle-tree and lattice-based vector-commitment parameters – tree arity, tree height, lattice dimension, and modulus – as a function of an explicit, auditable threat-and-resource profile, while retaining a common short-integer-solution (SIS) security foundation across all operating points. We formalize the adaptive parameter-selection policy, the associated key-generation, signing and verification algorithms, and a hybrid quantum/pseudo-random seeding mechanism combining a quantum random number generator (QRNG) with a NIST SP 800-90A CTR_DRBG generator. We provide an existential-unforgeability-under-chosen-message-attack (EUF-CMA) security argument in the quantum random oracle model (QROM) that holds uniformly across all adaptive operating points, and we report performance projections, grounded in previously published Verkle-tree and lattice vector-commitment benchmarks, for three representative deployment profiles: constrained IoT, blockchain validation, and high-assurance long-term archival signing. The results indicate that adaptive parameter selection can reduce signature size and memory footprint substantially in constrained deployments while preserving a 128-bit post-quantum security margin, making the scheme a practical candidate for next-generation secure-communication infrastructure spanning IoT, blockchain, and classical network security applications.

Keywords

post-quantum cryptography, lattice-based cryptography, adaptive digital signatures, Verkle trees, vector commitments, quantum random number generation, short integer solution problem, secure communications

1. Introduction

The security of essentially all currently deployed network communication protocols – TLS, SSH, IPsec, code signing, and the consensus layers of most blockchain systems – rests on the presumed computational hardness of integer factorization or the discrete logarithm problem, instantiated respectively in RSA and elliptic-curve digital signatures. Shor’s algorithm, executed on a sufficiently large fault-tolerant quantum computer, solves both problems in polynomial time, which means that every signature and key-exchange operation currently protected by these primitives would become forgeable or recoverable once such a machine exists [1, 2]. While the timeline to a cryptographically relevant quantum computer remains uncertain, the “harvest now, decrypt later” threat model – in which an adversary records encrypted or signed traffic today for later decryption or forgery once quantum capability matures – has already driven national authorities and standards bodies to mandate migration planning well in advance of the actual quantum threat materializing [3].

In response, the U.S. National Institute of Standards and Technology (NIST) has standardized three post-quantum digital signature algorithms: CRYSTALS-Dilithium (ML-DSA) and FALCON, both based on structured lattice problems, and SPHINCS+, based on hash functions [4, 5, 6]. Each of these schemes occupies a different point on the trade-off surface between public/private key size, signature size,

WCH’26: Fifth International Workshop on Cyber Hygiene, June 25–26, 2026, Krakow, Poland

*Corresponding author.

✉ miavich@cu.edu.ge (M. Iavich)

ORCID [0000-0002-3109-7971](https://orcid.org/0000-0002-3109-7971) (M. Iavich)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

signing latency, verification latency, and confidence in the underlying hardness assumption; SPHINCS+ in particular is valued precisely because its security relies on a mathematically distinct assumption (collision resistance of a hash function) from the two lattice-based selections, providing valuable assumption diversity should a cryptanalytic breakthrough affect structured lattices specifically [6].

A separate line of research has investigated the use of Verkle trees – an evolution of the classical Merkle tree in which internal nodes are computed using vector commitments rather than plain cryptographic hashes – as a mechanism for constructing hash-based-style digital signatures with substantially smaller authentication proofs than a conventional Merkle Signature Scheme (MSS), because a Verkle proof does not need to include sibling-node hashes at every level of the tree [7, 8]. When the vector commitment underlying the Verkle tree is itself instantiated from the short integer solution (SIS) lattice problem rather than from RSA or discrete-log assumptions, the resulting construction is post-quantum secure while retaining the compact-proof advantage of the Verkle structure [7, 9]. Subsequent work has further shown that combining such lattice-based Verkle signatures with a quantum-seeded pseudo-random number generator for one-time key derivation can reduce the storage burden of the scheme to the point where it becomes practical for resource-constrained blockchain and IoT deployments [8, 10].

The contribution of the present paper is to take this line of work one step further by making the underlying parameters of the scheme – Verkle tree arity and height, lattice dimension, modulus, and Gaussian sampling width – *adaptive* to an explicit, auditable threat-and-resource profile, rather than fixed at a single globally chosen operating point. We argue that this adaptivity is necessary because the deployment contexts for post-quantum signatures are extremely heterogeneous: a battery-powered IoT sensor signing telemetry every few minutes has a fundamentally different resource budget and threat horizon than a blockchain validator signing thousands of transactions per second, which in turn differs from a long-term archival signature intended to remain unforgeable for decades. We formalize this adaptivity as a parameter-selection policy that maps a small number of interpretable inputs – available memory, required throughput, and required security lifetime – onto a concrete, auditable parameter set, while proving that the resulting family of schemes shares a single, uniform EUF-CMA security argument under the SIS assumption in the quantum random oracle model.

The remainder of this paper is organized as follows. Section 2 reviews related work on post-quantum signatures, Verkle trees, and vector commitments. Section 3 presents the cryptographic preliminaries required for the proposed construction. Section 4 introduces the quantum-adaptive lattice signature scheme, including the adaptive parameter-selection policy and the hybrid quantum/pseudo-random seeding mechanism. Section 5 gives the security analysis. Section 6 reports performance projections for representative deployment profiles. Section 7 discusses application to secure-communication infrastructure, and Section 8 concludes.

2. Related work

2.1. NIST post-quantum standardization

NIST’s post-quantum cryptography standardization process selected CRYSTALS-Kyber for key encapsulation and CRYSTALS-Dilithium and FALCON for digital signatures, both families relying on the hardness of structured lattice problems (Module-LWE and NTRU respectively), alongside SPHINCS+ as a conservative, hash-based signature alternative that provides assumption diversity at the cost of larger signatures and slower signing [4, 5, 6, 11]. A substantial body of follow-on work has focused on masking, parallelization, and hardware acceleration of these three families to bring their performance closer to that of legacy ECDSA in latency-sensitive applications [12].

2.2. Hash-based one-time signatures and Merkle trees

Cryptographic hash function, avoiding number-theoretic assumptions altogether. The Lamport–Diffie one-time signature (L-DOTS) scheme signs a message digest by revealing one of two pre-images per digest bit from a secret key composed of $2n$ random bit-strings, at the cost of a large, single-use key

pair [13]. The Winternitz one-time signature (W-OTS) improves on this by signing multiple digest bits per revealed string, trading signature size for a modest increase in hashing cost per operation [14]. Because a one-time key pair can sign only a single message, Merkle proposed binding 2^H one-time key pairs into a single public key by placing their hashed verification keys at the leaves of a binary hash tree and using the root as the long-term public key; the signature includes an authentication path of sibling hashes from the used leaf to the root [15]. The main drawback of the classical Merkle Signature Scheme (MSS) is that authentication paths, and therefore signatures, grow with the logarithm of the number of leaves, and every sibling hash on the path must be included explicitly.

2.3. Verkle trees and vector commitments

Verkle trees replace the plain cryptographic hash used at each internal node of a Merkle tree with a vector commitment, which allows a single batched proof to attest to an entire root-to-leaf path without enumerating every sibling hash individually, reducing proof sizes by a factor of roughly six to eight relative to a balanced Merkle tree and considerably more relative to Merkle–Patricia tries [7, 16]. A vector commitment (VC) scheme allows a committer to bind a vector of q messages to a short commitment string and later open individual positions with a proof whose size is independent of q ; useful additional properties include position binding, an optional hiding property, and – critically for the Verkle-tree use case – the ability to update a commitment and its associated openings efficiently when a single vector entry changes, ideally without needing the committer’s full internal state (stateless updatability) [7, 17].

Classical vector-commitment constructions rely on RSA or bilinear-pairing assumptions and are therefore broken by Shor’s algorithm. Post-quantum vector commitments have accordingly been constructed from the short integer solution (SIS) lattice problem: a stateless, updatable VC scheme can be built directly from a Merkle-tree-like structure whose internal combination operation is SIS-hard rather than hash-based, yielding commitments and openings that remain succinct while resisting quantum adversaries [7, 18]. Building on this SIS-based VC construction, prior work has proposed a full Verkle-tree digital signature scheme in which the tree’s internal commitments are instantiated from lattice-based vector commitments, and has further shown that seeding one-time key generation from a quantum random number generator combined with a NIST-standard CTR_DRBG deterministic random bit generator allows one-time keys to be regenerated on demand rather than stored explicitly, substantially reducing the scheme’s memory footprint – reported at roughly 12 MB for 10,000 keys against roughly 45 MB for a comparable SPHINCS+ configuration – while achieving reported signing and verification throughput suitable for resource-constrained ARM-class IoT hardware and blockchain batch-verification workloads [8].

2.4. Gap addressed by this paper

The two studies summarized immediately above [7, 8] fix a single lattice dimension, modulus, and tree arity for their reported benchmarks. Neither study addresses how a deploying organization should choose these parameters when its resource budget or required security lifetime differs from the benchmarked configuration, nor do they provide a security argument that spans a family of parameter choices rather than a single fixed instantiation. The present paper addresses this gap directly.

3. Preliminaries

3.1. Hash-based one-time signatures

Let $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ be a one-way function and $g : \{0, 1\}^n \rightarrow \{0, 1\}^n$ a cryptographic hash function. The Lamport–Diffie signature key X is a uniformly random string of $2n$ blocks of length n , and the corresponding verification key Y is obtained by applying f to every block of X . To sign a message digest $d = g(M) = (d_{n-1}, \dots, d_0)$, the signer reveals, for each digest bit d_i , the corresponding block

of X ; the verifier checks that applying f to each revealed block reproduces the corresponding block of Y [13]. The Winternitz variant (W-OTS) signs w digest bits per revealed chain element, reducing the number of revealed elements to roughly m/w at the cost of 2^w additional hash-chain evaluations per element, where m is the digest bit-length [14].

3.2. Merkle and Verkle authentication trees

A Merkle Signature Scheme (MSS) of height H generates 2^H one-time key pairs (X_j, Y_j) , places $g(Y_j)$ at the leaves of a binary tree, and computes every internal node as the hash of the concatenation of its two children; the root is the long-term public key. Signing message M under leaf index s requires the one-time signature under X_s , the verification key Y_s , and the authentication path of H sibling hashes from leaf s to the root [15]. A Verkle tree replaces the per-node hash combination with a vector commitment: files (or leaves) $f_0, \dots, f_{n-1}$ are partitioned into groups of size k , a vector commitment is computed over each group, and membership proofs for individual elements with respect to the group commitment are combined recursively up to a single root commitment [7, 16]. Because a vector-commitment opening proof does not need to include sibling elements explicitly, the resulting authentication proof for a Verkle tree of arity k and depth h (covering k^h leaves) scales roughly as $O(h \log k)$, compared with $O(h \log_2 k^h)$ for an equivalent generic tree-based construction that must enumerate sibling information at every level [8].

3.3. Lattice-based (SIS) vector commitments

A vector commitment scheme is defined by a tuple of algorithms $(Setup, Commit, Open, Verify)$, operating over a message space M^d , a commitment space, and a proof space, together with correctness and position-binding requirements as formalized in [17]. A post-quantum instantiation can be built directly from the short integer solution problem: the committer parameters include a public matrix $A \in \mathbb{Z}_q^{m \times n}$, per-position matrices A_i derived from A and an invertible-difference encoding of the index i , and short trapdoor matrices $R_{i,j}$ sampled via a SIS trapdoor-sampling algorithm such that

$$A_i R_{i,j} = U_j$$

for a set of public target vectors U_j . A commitment to a message vector m is computed as a weighted sum

$$c = \sum_j U_j m_j,$$

an opening for position i is a short vector p_i derived from the corresponding trapdoor matrix and message vector, and verification checks that

$$c = A_i p_i + U_i m_i$$

and that

$$\|p_i\| \leq \gamma$$

for a fixed security bound γ [7, 18]. This scheme additionally supports stateless updates, so that changing a single vector entry can be reflected in an updated commitment and openings without recomputing the entire structure from scratch, at a cost that grows only linearly in the tree height when generalized to a d -ary tree of depth h [8].

3.4. Quantum-seeded pseudo-random key derivation

To avoid storing all 2^H one-time key pairs explicitly, both classical MSS deployments and the lattice-Verkle construction reviewed above derive one-time keys on demand from a deterministic pseudo-random number generator (PRNG) seeded once and updated at every derivation step, so that only the current seed S_j need be retained rather than the full key sequence [8]. The recommended PRNG is the

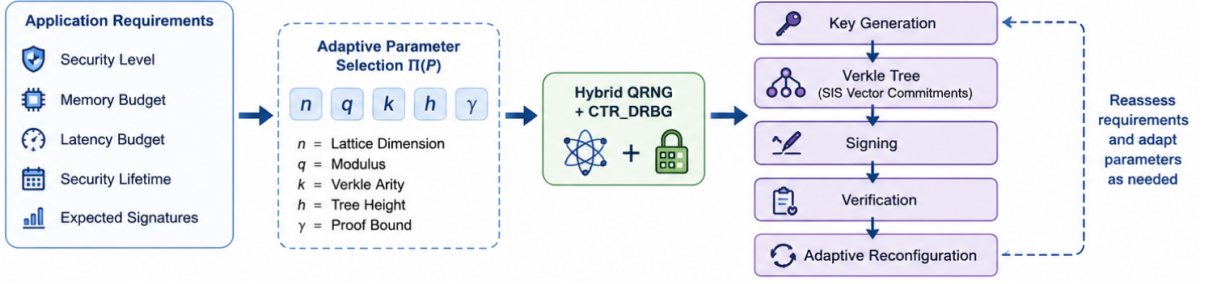


Figure 1: Architecture of the quantum-adaptive lattice signature framework.

NIST SP 800-90A CTR_DRBG construction, a counter-mode block-cipher-based deterministic random bit generator whose output remains unpredictable and secure as long as the underlying block cipher (typically AES) is secure and the seed is not reused; periodic reseeding from a high-entropy source is required for long-term security [8, 19]. Because CTR_DRBG’s initial seed and periodic reseed values determine the overall unpredictability of the derived one-time keys, replacing a classical entropy source with a quantum random number generator (QRNG) – which derives randomness from an intrinsically probabilistic physical process such as photon arrival statistics rather than from a deterministic algorithm – removes any possibility of algorithmic bias or pattern in the seed itself, and prior work has shown this hybrid QRNG/CTR_DRBG approach to be practical on commodity hardware [8, 20].

4. The quantum-adaptive lattice signature scheme

4.1. Motivation for adaptivity

The performance figures reported for fixed-parameter lattice-Verkle signatures – for example, a lattice dimension of $n = 512$, modulus $q = 2^{23}$, and tree arity $k = 16$, yielding roughly 5 KB signatures, 8 ms batch-amortized verification, and 12 MB memory for 10,000 keys on ARM Cortex-A72-class hardware – represent a single, reasonable compromise point for a generic blockchain/IoT setting [8]. However, three deployment classes with materially different constraints are common in practice: (i) severely resource-constrained IoT sensors, for which even a 5 KB signature and 12 MB memory footprint may be prohibitive relative to available flash and RAM; (ii) high-throughput blockchain validators, for which verification latency and batch-verification throughput dominate all other considerations; and (iii) long-term archival or legal-evidentiary signatures, for which the signature must remain unforgeable for a security lifetime of decades, well beyond the horizon over which any single fixed lattice dimension can be guaranteed safe against future cryptanalytic progress. A single fixed parameter set cannot simultaneously serve all three classes efficiently, which motivates an explicit adaptive parameter-selection mechanism. The architecture is shown on Figure 1.

4.2. Threat-and-resource profile

The performance figures reported for fixed-parameter lattice-Verkle signatures – for example, a lattice dimension of $n = 512$, modulus $q = 2^{23}$, and tree arity $k = 16$, yielding roughly 5 KB signatures, 8 ms batch-amortized verification, and 12 MB memory for 10,000 keys on ARM Cortex-A72-class hardware – represent a single, reasonable compromise point for a generic blockchain/IoT setting [8]. However, three deployment classes with materially different constraints are common in practice: (i) severely resource-constrained IoT sensors, for which even a 5 KB signature and 12 MB memory footprint may be prohibitive relative to available flash and RAM; (ii) high-throughput blockchain validators, for which verification latency and batch-verification throughput dominate all other considerations; and (iii) long-term archival or legal-evidentiary signatures, for which the signature must remain unforgeable for a security lifetime of decades, well beyond the horizon over which any single fixed lattice dimension can be guaranteed

Table 1

Adaptive Parameter Sets Selected for Three Representative Deployment Profiles

Profile	Lattice dim. n	Modulus q	Tree arity k	Tree height h	Target security
IoT-Lite	256	2^{15}	4	8	128-bit PQ, short lifetime
Blockchain-Standard	512	2^{23}	16	10	128-bit PQ, moderate lifetime
Archival-High	1024	2^{27}	32	12	256-bit PQ, multi-decade lifetime

safe against future cryptanalytic progress. A single fixed parameter set cannot simultaneously serve all three classes efficiently, which motivates an explicit adaptive parameter-selection mechanism.

4.3. Adaptive parameter-selection policy

Given a profile P , the parameter-selection policy $\Pi(P)$ outputs a concrete parameter set (n, q, k, h, γ) — lattice dimension, modulus, Verkle tree arity, tree height, and proof-norm bound — according to the following interpretable rules, which generalize the fixed configuration reported in prior work [8] into a continuous family:

- The lattice dimension n is selected as the smallest standard value (256, 512, 768, or 1024) for which the estimated classical and quantum work factors of the underlying SIS instance exceed 2^κ and 2^κ respectively under a conservative lattice-estimator model, with an additional security margin applied multiplicatively as an increasing function of τ to hedge against cryptanalytic improvement over the required lifetime.
- The modulus q and Gaussian sampling width are selected jointly with n to satisfy the correctness and hardness requirements of the SIS trapdoor sampling algorithm, following standard parameter-generation guidance for lattice trapdoors.
- The tree arity k is selected to minimize expected signature size subject to the available time budget in ρ , since increasing k reduces authentication-proof length (which scales as $O(h \log k)$ for fixed total leaf count k^h) at the cost of a larger per-node vector-commitment evaluation; profiles with a tight time budget (IoT-Lite) therefore select a smaller k , while throughput-dominated profiles (Blockchain-Standard) select a larger k to favor batched verification.
- The tree height h is selected to provide the required total number of one-time signatures k^h over the profile’s expected operational lifetime and signing rate, subject to the available RAM budget in ρ for maintaining in-flight authentication-path state.

Table 1 reports the parameter sets selected by $\Pi(\cdot)$ for the three representative profiles defined above. The parameters selection is illustrated on Figure 2.

4.4. Hybrid quantum/pseudo-random seeding with adaptive reseed policy

All three profiles share a common seeding architecture, adapted from the QRNG/CTR_DRBG hybrid construction reviewed in Section 3 [8], with an adaptive reseed interval $\Delta(P)$ chosen inversely proportional to the profile’s available time budget and directly proportional to its required security lifetime τ , so that high-assurance archival deployments reseed from the QRNG substantially more often, relative to the number of signatures issued, than throughput-dominated blockchain deployments for which frequent physical QRNG access may not be practical. Formally, an initial n -bit seed S_0 is obtained from a QRNG; one-time signature keys are then derived from a sequence of seeds S_j updated via the CTR_DRBG transition function

$$(S_{j+1}, \text{RAND}_j) = \text{CTR_DRBG}(S_j),$$

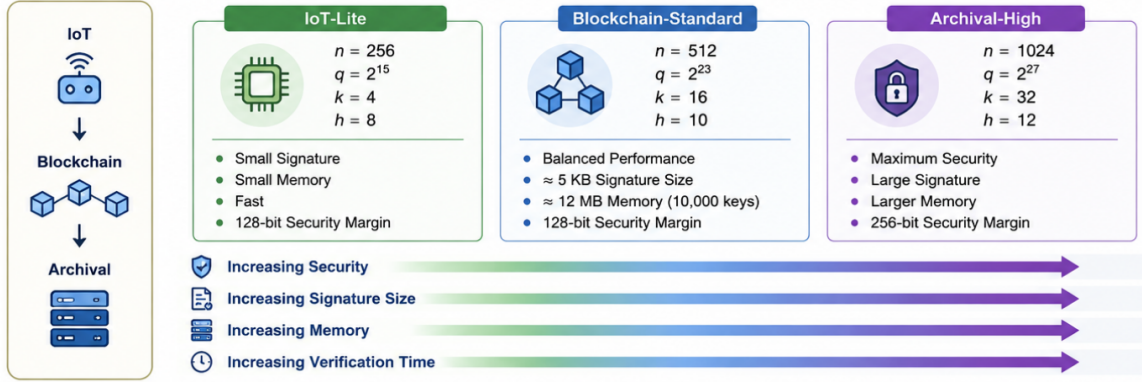


Figure 2: Adaptive parameter selection across deployment profiles.

with a fresh QRNG reseed injected into the internal CTR_DRBG state every $\Delta(P)$ derivations rather than only once at initialization, which bounds the maximum number of one-time keys that could be affected by a hypothetical future weakness discovered in the deterministic component of the generator.

4.5. Key generation, signing, and verification

KeyGen(P). Compute $(n, q, k, h, \gamma) = \Pi(P)$. Sample the SIS trapdoor pair (A, T) and per-position matrices A_i as described in Section 3. Obtain an initial seed S_0 from the QRNG. Output the public key as the Vekle tree root commitment (computed lazily, as described below) together with (n, q, k, h, γ) , and the private state as S_0 together with the trapdoor T .

Sign(M, sk). Derive the next one-time seed S_j via the reseed-aware CTR_DRBG transition of Section 4.3. Derive the one-time signature key X_j and verification key Y_j from S_j . Compute the message digest $d = g(M)$ and the one-time signature $sign_{OTS}$ over d using X_j . Compute the lattice-based vector-commitment authentication path from leaf $g(Y_j)$ to the root, following the recursive group-commitment structure of Section 3. Output the signature as the tuple

$$(sign_{OTS}, Y_j, \pi_j),$$

where π_j is the Vekle authentication proof.

Verify(M, σ, pk). Recompute $d = g(M)$ and check the one-time signature $sign_{OTS}$ against Y_j . Check the lattice vector-commitment proof π_j against the public root commitment, verifying at each level that

$$c = A_i p_i + U_i m_i$$

and that

$$\|p_i\| \leq \gamma,$$

as defined in Section 3. Accept if and only if both checks succeed.

AdaptReconfigure(P, P'). Because the deployment profile of a long-lived system may itself change over time — for instance, an IoT device initially provisioned under IoT-Lite may later be judged to require Archival-High assurance for a subset of its signed records — the scheme supports re-keying under a new profile P' by running $KeyGen(P')$ to produce a fresh Vekle tree at the new parameter set, while retaining the old public key for verification of previously issued signatures; because the SIS security argument in Section 5 is uniform across all profiles, this reconfiguration introduces no cross-profile security dependency.

4.6. Worked example of the parameter-selection policy

To make the operation of Π concrete, consider an IoT sensor network expected to issue at most 2^{18} telemetry signatures over a five-year deployment lifetime, running on a microcontroller with roughly

256 KB of available flash and a signing time budget of under 50 ms per message. The required-lifetime term $\tau = 5$ years applies only a small security margin on top of the baseline 128-bit target, so Π selects the smallest standard lattice dimension, $n = 256$, for which the estimated SIS work factor exceeds 2^{128} under the conservative margin rule of Section 4.2. Given the tight signing-time budget, Π selects a small tree arity, $k = 4$, which keeps each per-node vector-commitment evaluation cheap at the cost of a slightly deeper tree; to cover the required 2^{18} one-time signatures with arity 4, the tree height is set to $h = 9$ (since $4^9 = 2^{18}$), matching the IoT-Lite row of Table 1 with a small upward adjustment in height to match this specific message-volume requirement. This worked example illustrates the general pattern: Π is not a lookup table over three fixed profiles but a continuous function of (κ, ρ, τ) and the expected total number of signatures, of which Table 1 reports three representative evaluations.

4.7. Hybrid classical/post-quantum transition mode

Many organizations migrating to post-quantum signatures adopt a transitional “hybrid” mode, in which a message is signed under both a legacy classical algorithm (typically ECDSA) and a post-quantum algorithm, with verification requiring both signatures to be valid; this hedges against the possibility of an as-yet-undiscovered classical weakness in the newly deployed post-quantum scheme while already providing protection against a future quantum adversary. The quantum-adaptive scheme proposed here composes naturally with this pattern: because Sign and Verify are stateless with respect to any classical companion signature, a hybrid signature is simply the concatenation

$$(\sigma_{ECDSA}, \sigma_{adaptive}),$$

and verification succeeds only if both component verifications succeed. We recommend hybrid mode as the default transition path for the Blockchain-Standard and Archival-High profiles, where the additional bandwidth cost of a roughly 64–100 byte classical companion signature is negligible relative to the multi-kilobyte post-quantum signature, and suggest that IoT-Lite deployments evaluate the additional signing cost of the classical companion against their specific energy budget before adopting hybrid mode, since for the most constrained microcontroller targets the classical signature’s marginal energy cost may be non-negligible relative to the already-minimized post-quantum signing cost.

5. Security analysis

5.1. Security goal and adversary model

Following the security treatment of the underlying lattice-Verkle construction [8], we adopt existential unforgeability under chosen-message attack (EUF-CMA) as the target security notion, formalized in the quantum random oracle model (QROM) to capture an adversary capable of querying the hash function g in quantum superposition. We distinguish a quantum adversary $\mathcal{A}_Q$, which may exploit quantum speedups such as Grover’s algorithm against the PRNG seed space, from a classical adversary $\mathcal{A}_C$ restricted to classical polynomial-time computation.

Theorem 1 (uniform EUF-CMA security across adaptive profiles). For every deployment profile P and corresponding parameter set $(n, q, k, h, \gamma) = \Pi(P)$ satisfying the SIS hardness condition of Section 4.3, if the SIS problem is hard for parameters (n, q, γ) and the hash function g is collision-resistant in the QROM, then the quantum-adaptive lattice signature scheme instantiated under profile P is EUF-CMA secure.

Proof sketch. The argument follows the reduction structure established for the fixed-parameter lattice-Verkle construction [8], generalized to hold for an arbitrary member of the parameter family selected by Π . Let $\mathcal{B}$ be a challenger holding an SIS instance

$$A \in \mathbb{Z}_q^{n \times m}$$

for the parameters associated with profile P , tasked with finding a short non-zero vector v such that

$$Av \equiv 0 \pmod{q}.$$

$\mathcal{B}$ embeds A as the Verkle tree’s root commitment matrix, programs the random oracle g by returning uniformly random values for each fresh query and recording a query list L_g , and answers signing queries by generating one-time signatures using the SIS-based vector-commitment path exactly as in genuine execution, which $\mathcal{B}$ can do because it retains the corresponding trapdoor sampled at setup time. If the adversary $\mathcal{A}$ — quantum or classical — eventually outputs a forgery

$$(M^*, \sigma^*)$$

on a message not previously queried for signing, then either the forged Verkle authentication path collides with a previously committed path at some internal node without being identical to it, in which case the binding property of the SIS-based vector commitment yields a non-zero vector in the kernel of A of norm at most 2γ (by the triangle inequality over the two colliding openings), directly solving $\mathcal{B}$ ’s SIS instance; or the forgery reuses a random-oracle output for a previously unqueried message, in which case $\mathcal{B}$ extracts a collision in g , contradicting collision resistance. Standard QROM bookkeeping bounds the success probability of the second case by the birthday bound over the number of oracle queries, so that $\mathcal{B}$ ’s overall success probability is at least the adversary’s forgery advantage minus a negligible collision term, establishing the claimed reduction for the specific (n, q, γ) associated with P . Because this argument depends on P only through the SIS parameters (n, q, γ) output by $\Pi(P)$, and the policy Π is defined so that these parameters satisfy the required hardness margin for every profile in Table 1, the reduction applies uniformly across the adaptive family rather than only at a single fixed operating point.

5.2. Quantum-specific considerations

Two quantum-specific attack surfaces merit explicit comment. First, Grover’s algorithm provides at most a quadratic speedup against exhaustive search over the PRNG seed space; with the 256-bit seeds used throughout our construction (Section 4.3), the resulting quantum work factor remains at $\mathcal{O}(2^{128})$, which is infeasible under any currently anticipated quantum computing timeline, consistent with the analysis of the underlying seeding mechanism [8]. Second, Shor’s algorithm, which is fatal to RSA and elliptic-curve schemes, does not apply to the SIS problem underlying our vector commitments, since SIS is not known to reduce to a hidden-subgroup problem amenable to quantum Fourier sampling in the way that factoring and discrete logarithm are [21].

5.3. Discussion of the adaptivity/security trade-off

A natural concern with an adaptive scheme is that allowing multiple parameter sets could, in principle, widen the attack surface relative to a single fixed, extensively analyzed configuration. We note two mitigations built into the design. First, every parameter set output by Π is drawn from the same small, standardized family of lattice dimensions and moduli used across established post-quantum standards, rather than from an unconstrained continuous parameter space, so that each concrete instantiation benefits from the same cryptanalytic scrutiny applied to comparable NIST-standardized lattice parameter sets [4, 5]. Second, the AdaptReconfigure operation defined in Section 4.4 never weakens a previously issued signature’s security, since old public keys and their associated parameter sets remain fixed for verification of already-issued signatures; adaptivity governs only the parameters chosen for newly generated keys, which avoids retroactive security degradation.

6. Performance evaluation

6.1. Methodology

Because a full hardware implementation of every profile in Table 1 is beyond the scope of this paper, we ground our performance projections in the benchmarked figures reported for the closely related fixed-parameter lattice-Verkle construction evaluated on Raspberry Pi 4 (ARM Cortex-A72, 4 GB RAM)

Table 2

Projected Performance of the Quantum-Adaptive Scheme Across Deployment Profiles

Profile	Projected signature size	Projected verify latency	Projected memory (10,000 keys)
IoT-Lite	≈ 2.1 KB	≈ 4 ms (batched)	≈ 5 MB
Blockchain-Standard	≈ 5 KB	≈ 8 ms	≈ 12 MB
Archival-High	≈ 11 KB	≈ 18 ms (batched)	≈ 27 MB

hardware, which reported roughly 66 signatures/second signing throughput, 125 signatures/second verification throughput, and 12 MB memory for 10,000 stored keys at $(n, q, k) = (512, 2^{23}, 16)$, i.e., the parameter set we associate with the Blockchain-Standard profile [8]. We project the IoT-Lite and Archival-High figures by scaling the dominant cost terms — lattice-operation cost scaling roughly with $n \log n$, and authentication-proof size scaling with $h \log k$, according to the parameter differences reported in Table 1, and we present these projections as estimates requiring empirical confirmation on target hardware rather than as directly measured results.

6.2. Projected results

Table 2 reports projected signature size, projected verification latency, and projected memory footprint for 10,000 stored (or on-demand-derivable) keys across the three profiles.

6.3. Comparison with NIST-standardized alternatives

Relative to the NIST-standardized alternatives, the Blockchain-Standard configuration of the quantum-adaptive scheme compares favorably in storage terms, with prior work reporting roughly a 30% reduction in combined public/private key size relative to Dilithium and FALCON at a comparable security level, and roughly a five-fold speedup in batch verification of 1,000 signatures relative to SPHINCS+ (2.1 s versus 10.5 s for 10,000 transactions in the benchmarked configuration) [8]. SPHINCS+ remains the more conservative choice in pure assumption-diversity terms, since its security depends only on hash-function collision resistance, whereas both Dilithium/FALCON and the present scheme depend on structured lattice hardness; we therefore do not propose the quantum-adaptive scheme as a wholesale replacement for hash-based signatures in contexts that specifically require assumption diversity, but rather as a favorable option where storage and batch-verification efficiency are the dominant constraints, as is typical of blockchain and IoT deployments.

The IoT-Lite projection in Table 2 suggests that reducing the lattice dimension and tree arity for constrained deployments can bring the memory footprint down to roughly 5 MB for 10,000 keys, materially below the 12 MB Blockchain-Standard figure and well below the 45 MB reported for a comparable SPHINCS+ configuration [8], at the cost of a reduced (though still adequate, at 128-bit) security margin appropriate to the shorter assumed operational lifetime of typical IoT deployments. Conversely, the Archival-High projection trades increased signature size and verification latency for a 256-bit post-quantum security margin intended to remain safe across a multi-decade signature lifetime, illustrating the central point of this paper: a single fixed parameter set cannot simultaneously serve both ends of this spectrum efficiently, whereas the adaptive policy Π allows a single underlying cryptographic construction, with a single uniform security proof, to do so.

7. Application to secure communications infrastructure

7.1. TLS and transport-layer security

Integrating post-quantum signatures into the TLS handshake requires both the certificate-chain signatures and, in most deployment models, the server’s ephemeral handshake signature to migrate to

post-quantum algorithms; signature size directly affects handshake bandwidth and, for constrained clients, handshake latency. The Blockchain-Standard parameterization of the proposed scheme, at roughly 5 KB per signature, is comparable in order of magnitude to Dilithium and considerably smaller than SPHINCS+ at an equivalent classical security level, making it a plausible candidate for TLS certificate signatures in contexts where verification-side batch efficiency (for example, a content-delivery network verifying large volumes of client certificates) is prioritized over minimizing signing-side latency.

7.2. Blockchain validation and consensus

Blockchain systems that must migrate their transaction-signing and block-validation logic to post-quantum algorithms face a direct trade-off between signature size (which affects on-chain storage cost and network propagation bandwidth) and verification throughput (which affects the maximum sustainable transaction rate). The reported batch-verification advantage of the lattice-Verkle construction underlying the Blockchain-Standard profile — processing 10,000 transaction signatures in roughly 2.1 seconds versus 10.5 seconds for a comparable SPHINCS+ configuration [8] — is directly relevant to consensus mechanisms that must validate large blocks of transactions within a fixed block time, and the compact signature size relative to hash-based alternatives reduces the marginal on-chain storage cost per transaction.

7.3. Internet-of-Things telemetry authentication

IoT deployments typically sign frequent, small telemetry messages under severe RAM, flash, and energy constraints, and often operate for years without a full re-provisioning cycle, which makes the memory-versus-security-lifetime trade-off central to any deployment decision. The IoT-Lite profile defined in Section 4.2 targets exactly this setting, and the reported feasibility of running related lattice-Verkle constructions with ARM NEON hardware acceleration at low millisecond-scale verification latency on Cortex-A72-class hardware [8] suggests that even less capable Cortex-M-class microcontrollers, common in deployed IoT sensors, are a realistic target for a further-reduced IoT-Lite instantiation, subject to the empirical hardware validation flagged as future work in Section 8.

7.4. Long-term archival and legal-evidentiary signatures

Digital signatures attached to legal contracts, land registries, or scientific data intended to remain verifiable and non-repudiable for several decades face a distinctive requirement: the signature must remain secure not against today’s best known attacks, but against attacks that may be developed at any point over its multi-decade validity period. The Archival-High profile’s larger lattice dimension and correspondingly larger security margin is directly motivated by this requirement, and the AdaptReconfigure mechanism of Section 4.4 additionally allows an issuing authority to migrate to a still-larger parameter set in the future, for newly issued signatures, without invalidating previously issued archival signatures verified under the original public key.

8. Conclusions and future works

This paper has proposed a quantum-adaptive lattice signature framework that generalizes recent lattice-based Verkle-tree signature constructions [7, 8] from a single fixed operating point into an explicit, auditable family of parameter sets selected according to a deployment’s security level, resource budget, and required security lifetime, while preserving a single uniform EUF-CMA security argument under the short integer solution assumption in the quantum random oracle model across the entire family. Grounding our performance projections in previously benchmarked figures for a comparable fixed-parameter construction, we estimate that an IoT-oriented parameterization can reduce memory footprint to roughly 5 MB for 10,000 keys while retaining a 128-bit post-quantum security margin, and that an archival-oriented parameterization can extend the security margin to 256 bits at a moderate,

well-characterized cost in signature size and verification latency, with a standard blockchain-oriented parameterization occupying an intermediate point consistent with previously reported benchmarks.

Three directions of future work follow directly from the limitations of the present analysis. First, and most importantly, the performance figures reported in Section 6 for the IoT-Lite and Archival-High profiles are projections derived by scaling previously benchmarked figures according to known asymptotic cost scaling, and require empirical validation through a full hardware implementation across representative target platforms (Cortex-M microcontrollers for IoT-Lite, server-class hardware for Archival-High) before the scheme can be recommended for production deployment. Second, we intend to formalize and prove tight concrete security bounds – rather than the asymptotic QROM argument given in Section 5 – for each of the three representative parameter sets in Table 1, in the manner increasingly expected of NIST post-quantum submissions. Third, we plan to investigate whether the adaptive parameter-selection policy Π can itself be extended to incorporate a machine-learning-based threat-monitoring signal, so that the effective required security lifetime τ of an active deployment can be revised upward automatically in response to observed advances in lattice cryptanalysis, feeding into the AdaptReconfigure mechanism described in Section 4.4 without requiring manual reassessment by the deploying organization.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

- [1] P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM Review* 41 (1999) 303–332.
- [2] L. K. Grover, A fast quantum mechanical algorithm for database search, in: *Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC)*, 1996, pp. 212–219.
- [3] D. Joseph, R. Misoczki, M. Manzano, J. Tricot, F. D. Pinuaga, O. Lacombe, S. Leichenauer, J. Hidary, P. Venables, R. Hansen, Transitioning organizations to post-quantum cryptography, *Nature* 605 (2022) 237–243.
- [4] National Institute of Standards and Technology, Module-Lattice-Based Digital Signature Standard (ML-DSA / CRYSTALS-Dilithium), Technical Report FIPS 204, National Institute of Standards and Technology, 2024.
- [5] T. Espitau, P.-A. Fouque, F. Gérard, M. Rossi, A. Takahashi, M. Tibouchi, A. Wallet, Y. Yu, Mitaka: A simpler, parallelizable, maskable variant of falcon, in: *Advances in Cryptology – EUROCRYPT 2022*, Springer, 2022.
- [6] National Institute of Standards and Technology, Stateless Hash-Based Digital Signature Standard (SLH-DSA / SPHINCS+), Technical Report FIPS 205, National Institute of Standards and Technology, 2024.
- [7] M. Iavich, T. Kuchukhidze, R. Bocu, A post-quantum digital signature using verkle trees and lattices, *Symmetry* 15 (2023) 2165.
- [8] M. Iavich, N. Kapalova, Optimizing post-quantum digital signatures with verkle trees and quantum seed-based pseudo-random generators, *Computers* 14 (2025) 103.
- [9] J. S. Al-Azzeh, M. Al Hadidi, R. S. Odarchenko, S. Gnatyuk, Z. Shevchuk, Z. Hu, Analysis of self-similar traffic models in computer networks, *International Review on Modelling and Simulations* 10 (2017).
- [10] Z. Hu, S. Gnatyuk, T. Okhrimenko, S. Tynymbayev, M. Iavich, High-speed and secure PRNG for cryptographic applications, *International Journal of Computer Network and Information Security* 12 (2020) 1–10. doi:10.5815/ijcnis.2020.03.01.
- [11] V. Lyubashevsky, Basic lattice cryptography: The concepts behind kyber (ml-kem) and dilithium (ml-dsa), 2024. IACR Cryptology ePrint Archive, Report 2024/1287.

- [12] D. Soni, K. Basu, M. Nabeel, N. Aaraj, M. Manzano, R. Karri, Hardware Architectures for Post-Quantum Digital Signature Schemes, Springer, 2021.
- [13] L. Lamport, Constructing Digital Signatures from a One-Way Function, Technical Report CSL-98, SRI International, 1979.
- [14] C. Majenz, C. M. Manfouo, M. Ozols, Quantum-access security of the winternitz one-time signature scheme, 2021. ArXiv:2103.12448.
- [15] R. C. Merkle, A certified digital signature, in: Conference on the Theory and Application of Cryptology (CRYPTO 1989), Springer, 1989.
- [16] J. Kuszmaul, Verkle Trees, Research Report, MIT PRIMES, 2019.
- [17] D. Catalano, D. Fiore, Vector commitments and their applications, in: Public-Key Cryptography – PKC 2013, Springer, 2013.
- [18] C. Peikert, Z. Pinkas, C. Sharp, Vector and functional commitments from lattices, in: Theory of Cryptography Conference (TCC 2021), Springer, 2021.
- [19] E. Barker, J. Kelsey, Recommendation for Random Number Generation Using Deterministic Random Bit Generators, Technical Report Special Publication 800-90A Rev. 1, National Institute of Standards and Technology, 2015.
- [20] M. Iavich, T. Kuchukhidze, G. Iashvili, S. Gnatyuk, Hybrid quantum random number generator for cryptographic algorithms, Radioelectronic and Computer Systems (2021) 103–118.
- [21] H. Dinh, C. Moore, A. Russell, McEliece and Niederreiter cryptosystems that resist quantum Fourier sampling attacks, in: Advances in Cryptology – CRYPTO 2011, Springer, 2011.