POLITECNICO DI TORINO
Repository ISTITUZIONALE

Hybrid Post-Quantum Cryptography and Ethereum Signatures: A Comprehensive Performance Benchmarking Study

Original
Hybrid Post-Quantum Cryptography and Ethereum Signatures: A Comprehensive Performance Benchmarking Study / Ahmed, M., Sisinni, S.. - ELETTRONICO. - 4198:(2026). (ITASEC & SERICS 2026 Joint National Conference on Cybersecurity 2026 Cagliari (ITA) February 9-13, 2026).

Availability:
This version is available at: 11583/3006648 since: 2026-01-16T13:05:51Z

Publisher:
CEUR-WS

Published
DOI:

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

26 November 2026

Hybrid Post-Quantum Cryptography and Ethereum Signatures: A Comprehensive Performance Benchmarking Study

Mishaal Ahmed[1,*,†], Silvia Sisinni[2,†]

[1]National Research Council (CNR)–ISASI, Napoli 80078, Italy
[2]Politecnico di Torino, DAUIN, Torino 10129, Italy

Abstract

The emergence of Quantum computing has threatened the Elliptic Curve Cryptography (ECC) used by blockchains such as Ethereum. In this paper, we provide a performance benchmarking of a hybrid signature architecture combining Ethereum's built-in ECDSA with NIST-standardized post-quantum cryptography (PQC) signature algorithms. We examine and compare three PQC algorithms; CRYSTALS-Dilithium, FALCON, and SPHINCS$^+$ across different security levels by integrating PQC signatures into transaction data while preserving ECDSA for transaction authentication. We provide metrics such as key generation, blockchain gas prices, key and signature sizes, and signing and verification in our analysis. The experiments demonstrate key generation times between 8.78 ms and 133.60 ms, signature timings between 6.14 ms and 1675.18 ms, and verification times between 2.82 ms and 8.29 ms, representing an acceptable overhead as compared to ECDSA. The hybrid approach provides a workable and backward-compatible migration path toward quantum-resistant blockchain systems, despite the fact that PQC keys and signatures are significantly larger, they require about 87K and 1,740K gas units for PQC key registration, which is two to eighty-seven times more expensive than standard ECDSA transactions. Finally, we draw attention to the trade-offs in hybrid PQC blockchain systems between cost, security, and performance.

Keywords

Post Quantum Cryptography (PQC), Elliptic Curve Cryptography (ECC), Ethereum, Hybrid Signatures

1. Introduction

The dynamics of digital infrastructure have been drastically changed by blockchain technology, which offers decentralized solutions in identity management, supply chain, finance, and many other domains [1, 2]. Cryptographic primitives are used by Ethereum, one of the most important blockchain platforms, for account security, transaction authentication, and smart contract execution [3, 4]. Currently, elliptic curve cryptography, more especially the `secp256k1` curve used in Elliptic Curve Digital Signature Algorithm (ECDSA), provides robust protection against classical computational attacks against blockchain systems. However, the fundamental principles of cryptography are in grave danger due to the quick development of quantum computing [5]. For instance, Shor's algorithm can break elliptic curve cryptography and other number-based theoretical problems that are at the heart of current blockchain security when running on a quantum computer with enough power. The cryptography community has agreed that it is needed to switch to quantum-resistant algorithms before we have large-scale quantum computers that can break ECDSA [6]. Even if the quantum computers are not ready now, we need to consider the attack vector "harvest now, decrypt later" attack scenario in which adversaries could accumulate encrypted data today and decrypt it later when quantum computers are available.

In 2022, several signature schemes were chosen because of the National Institute of Standards and Technology (NIST) post-quantum cryptography standardization effort [7]. The NIST-standardized signature schemes, each providing distinct security-performance trade-offs, include Module-Lattice-Based Digital

Joint National Conference on Cybersecurity (ITASEC & SERICS 2026), February 09-13, 2026, Cagliari, IT
*Corresponding author.
†These authors contributed equally.
mishaalahmed@cnr.it (M. Ahmed); silvia.sisinni@polito.it (S. Sisinni)
https://github.com/mishaalahmed (M. Ahmed)
0000-0003-2875-7438 (M. Ahmed); 0000-0003-1870-6303 (S. Sisinni)
© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073
published 2026-04-23

CEUR-WS.org/Vol-4198/paper64.pdf

Signature Algorithm (ML-DSA), Stateless Hash-based Digital Signature Algorithm (SLH-DSA), and FALCON. Nevertheless, the integration of blockchain systems with Post-Quantum Cryptographic (PQC) signature schemes exhibits numerous challenges. Some of the core issues are associated with performance overheads because of consensus and mining, gas costs involved with transactions, and the storage capacity of proofs.

Using hybrid or heuristic signature schemes that integrate classical and post-quantum cryptography is one workable way to counter the quantum threat in blockchains [8]. This approach signs standard transactions compatible with ECDSA and, in parallel, incorporates PQC signatures to achieve long-term quantum resilience. There are various benefits of this hybrid signature approach. First, using a PQC signature provides immediate resistance to quantum-enabled attacks. Second, it is fully backward compatible with Ethereum infrastructure. Third, it facilitates a gradual migration path by reducing the need for hard forks. Finally, the signer is able to modulate the balance between classical and quantum security, providing flexible adaptation to evolving threat models.

Such a hybrid approach preserves the functionality of the blockchain and secure it from emerging quantum threats. It maintains the validation of transactions for present consensus as it uses ECDSA signature. In addition, PQC signature provides extra cryptographic artifact which can be verified off-chain and can be used in future protocol revisions. It is valuable for blockchain systems where protocols are rapidly evolving.

This study aims to investigate the feasibility of integrating PQC signature schemes into Ethereum through a hybrid signature architecture (PQC-ECDSA) that combine PQC and Ethereum. We benchmark the computational performance, analyze storage, bandwidth overheads, and assess gas costs based on three NIST defined PQC algorithms and compare them with ECDSA baseline spanning key generation, signing, and verification procedures. We also analyze the trade-offs between the NIST security levels and performance characteristics to present algorithm selection decisions.

1.1. Contributions of the Paper

To the best of our knowledge, this study provides the first complete benchmark analysis of hybrid PQC combined Ethereum signatures. The key contributions are as follows:

- **Novel Hybrid Architecture:** We design and assess a novel hybrid signature architecture that utilizes standard ECDSA network-level transaction validity while embedding Post-Quantum (PQ) signatures directly within the transaction payload for off-chain verification, demonstrating a practical, application-layer approach to quantum resistance.
- **Comprehensive Gas Profiling:** We provide a comprehensive gas cost analysis to quantify the real-world feasibility of PQC integration on Ethereum, detailing explicit measurements for state-based key registration and the execution of transactions containing large embedded PQC payloads.
- **Baseline Comparative Benchmarking:** We evaluate the on-chain performance of recently recommended major NIST PQC signature algorithms, uniquely evaluating the on-chain gas economics and smart contract performance of embedded PQ transaction payloads against a standard ECDSA baseline, providing actionable insights to support developers in informed decision-making for quantum-resilient migration paths.
- **Backward-Compatible Migration Strategy:** We establish a practical transition framework for decentralized applications, demonstrating how developers can achieve application-layer post-quantum security immediately, without requiring underlying protocol-level hard forks or network-wide consensus changes. We also provide recommendation based on the experimental results.

1.2. Organization of the Paper

The rest of the paper organized in six main sections. Section 2 presents the preliminaries (background) on the concepts related to the presented study. These concepts include PQC, Ethereum Blockchain

Architecture, Hybrid Signatures, among others. Section 3 surveys the related work on blockchain security and quantum-resistant signatures. Section 4 describes the experimental methodology, including algorithm-selection criteria, the benchmarking framework, and the evaluation metrics used in this study. Section 5 reports the results of our analysis, followed in Section 6 by a detailed discussion of their implications and recommended design considerations. Finally, Section 7 concludes the paper and outlines directions for future research.

2. Preliminaries

This section provides fundamental knowledge of the concepts related to the research presented article. We start with providing a background on PQC. After that, we present an overview of the Ethereum blockchain architecture. Next, we introduce the concept of hybrid signatures. Subsequently, we introduce blockchain security and quantum threats. Finally, we discuss Ethereum gas model and costs.

2.1. Post-Quantum Cryptography

Post-quantum cryptography refers to cryptographic algorithms running on classical computers and designed to remain secure against adversaries equipped with both classical and quantum computers [9]. Unlike quantum cryptography, which relies on physical properties of quantum systems, PQC schemes derive their security from mathematical problems believed to be resistant to quantum attacks. In 2016, NIST launched its Post-Quantum Cryptography Standardization Process to evaluate candidate algorithms across several families, including multivariate, lattice-based, code-based, hash-based, and isogeny-based constructions [10, 11]. This effort culminated on 13 August 2024, when NIST issued the first set of finalized Federal Information Processing Standards (FIPS). These standards include FIPS 203 (ML-KEM, formerly known as CRYSTALS-Kyber) [12], FIPS 204 (ML-DSA, formerly CRYSTALS-Dilithium) [13], and FIPS 205 (SLH-DSA, formerly SPHINCS$^+$) [14], marking the formal adoption of the first generation of quantum-resistant public-key cryptographic primitives. In contrast, FALCON has not yet been approved as a FIPS standard; a draft standardization track is anticipated, but the algorithm remains non-standard at present.

The lattice-based signature scheme ML-DSA is based on Module Learning With Errors (MLWE) problem. There are three standardized parameter sets aligned with NIST security levels 2, 3, and 5. It offers a design that balances computational efficiency with compact signatures. As per NIST ML-DSA is standardized as the default digital signature scheme. It is mainly used for public-key authentication tasks [15]. ML-DSA is used in high volume use cases because of its performance characteristics. For instance, it is used in Transport Layer Security (TLS) authentication, certificate infrastructures, and blockchain protocols where verification throughput is critical [16].

NIST have also provided a stateless hash-based signature scheme, SLH-DSA. The security of this scheme relies mainly on the difficulty of finding the hash collisions. The design of this scheme is conservative to make it suitable for systems needing long-term assurance as it avoids algebraic structures that expose vulnerability. The size of its signature and key is significantly larger than lattice-base schemes however, it provides predictable security with minimum assumptions. This scheme is highly useful in environments in which durability and robustness outweigh performance and bandwidth considerations. Some examples include secure boot, archival signatures, high-assurance environments, and firmware and software distribution [17, 18, 19].

Similarly, another lattice-based signature scheme introduced by NIST is FALCON. It was constructed from NTRU family of hardness assumptions. It supports NIST security levels 1 and 5 and combines highly efficient signing operations with compact signatures. FALCON is best in applications such as embedded devices, communication systems, Internet of Things (IoT) networks, latency sensitive or bandwidth constrained deployments [20, 21, 22]. Although FALCON has a favorable performance characterstics but it has not been finalized as a FIPS standard. FALCON's adoption remains contingent in compliance sensitive settings.

Recently, a second set of PQC schemes have been proposed for standardization by NIST, providing alternative options with different performance and security trade-offs. This second set includes Hamming Quasic-Cyclic (HQC), a code based scheme complementing ML-KEM algorithm. The draft standardization is expected in 2026 and finalization in 2027. In addition, NIST is conducting a competition for additional PQC signature schemes to expand the diversity of secure algorithms suitable for diverse applications. This competition is based on PQ threshold signatures which is highly relevant to blockchain applications. Threshold signatures enable distributed signing, where a subset of participants jointly sign a transaction, enhancing fault tolerance and decentralization in blockchain networks. Integrating PQ threshold signatures with hybrid classical + PQC schemes could provide future proof security for blockchain systems [23].

2.2. Ethereum Blockchain Architecture

Ethereum is the world's leading in decentralized, open-source blockchain system that enables smart contracts and decentralization applications. It allows developers to deploy contracts that execute autonomously, without centralized control, which led to the adoption of Ethereum as a general purpose blockchain for financial, governance and data-sharing applications [24]. Ethereum uses an account-based model, in which externally owned accounts and smart contracts send transactions to one another. A transaction includes specify a sender, receiver. data payload and a transaction fee (gas), which is needed to run operations and store data on-chain. This computation is deterministic and verifiable through the Ethereum Virtual Machine (EVM), which executes all smart contracts across nodes.

Transactions are signed using the ECDSA, over the `secp256k1` curve, which offers strong security in classical computing environments [25]. However, this cryptographic mechanism is susceptible to possible quantum attacks, motivating research exploration of potential post-quantum and hybrid security solutions. Ethereum's roadmap prioritizes scalability, security and long-term sustainability via technologies like layer-2 networks and advanced cryptographic primitives [26]. Ethereum's ongoing efforts make it a perfect system for testing post-quantum and hybrid signature schemes in realistic smart contract runtime environments.

2.3. Hybrid Signatures

Hybrid signatures are cryptographic constructions that combine a PQ signature scheme with a classical signature scheme to generate, for the same message, one single coherent signature [27]. This ensures long-term security against classical and quantum attacks by ensuring that the message can still be validated even if one of the schemes is broken in future. Subsequently, hybrid signature approach allows systems to maintain backward compatibility while being prepared for those new attacks that might be expected in the future.

Despite the security advantages provided by hybrid signatures, they are not free of drawbacks [28]. For example, they increase the computational overhead. Since each message must be signed twice (once using a classical scheme and once using a PQ scheme). Moreover, hybrid signature approach produces large signatures. As a consequence, it potentially raises storage overhead and transaction fees in blockchain networks. Furthermore, the hybrid signature approach also increases the verification complexity. Due to the fact that both classical and PQ components of the scheme have to be validated. Consequently, this may affect performance in high-throughput or resource-limited environments.

2.4. Blockchain Security and Quantum Threats

In blockchain systems, cryptographic primitives underpin all security-critical operations. For instance, in Ethereum, for the security of account ownership, transaction authorization, and interactions with smart contracts is based on ECDSA [29]. Each transaction in Ethereum blockchain is signed with a private key which is associated with an account. It provides cryptographic proof of authorization and enable decentralized consensus on blockchain state transitions [30]. Blockchain systems are vulnerable to quantum threats because of the risk of production of large-scale quantum computers

[31]. Distributed ledgers used in blockchain systems are immutable; however, the transactions secured with quantum vulnerable primitives cannot be protected. Nevertheless, because of the rising risks related to blockchain and quantum computing, many approaches have been proposed to address the challenges [32].

One approach is to completely replace the classical cryptographic signature schemes with PQC schemes through hard-fork mechanism. However, it requires consensus of broad community and it may also introduce compatibility challenges. Another approach is to perform soft-fork on blockchains to enable incremental deployment. However, it may result into inconsistent security because it can create heterogeneous security guarantees during the transition period. Similarly, there is one more approach known as Hybrid signature approach which we have explored in this work. It is an appealing alternative and the best of both the worlds. Hybrid signature approach resilience against quantum threats as it introduces quantum resistance while preserving compatibility with existing infrastructure.

2.5. Ethereum Gas Model and Costs

The economic model provided by Ethereum blockchain involves the use of cryptocurrency also known as *gas*. EVM imposes explicit storage and computational costs because of its *gas accounting* model. For instance, it requires 20.000 gas per 32-byte word to write data to the contract storage. Similarly, any modification to the written data requires 5.000 gas. This gas cost is incurred because it adds burden to the on-chain state of the network. Moreover, it constitutes a significant factor in the design of cryptographic protocols implemented on Ethereum. In this work, we have represented the computing resources used by each operation by gas consumption metrics (represented in standard Ethereum gas units). To compute the monetary cost in Ethereum (ETH), the following formula can be utilized:

$$\text{Cost(ETH)} = (\text{Gas_Units} \times \text{Gas_Price}) \times 10^{-18}$$

As per the formula, the gas price is derived from the current gas price of the network's oracle which is generally represented in wei per gas unit. In contrast to classical signature schemes, the post quantum signature schemes involve substantially larger public keys and signatures. If we put them on-chain or transmit them within transactions, it significantly affects the gas consumption. As a result, it is necessary to precisely select PQC algorithm for blockchain systems to find a balance between security requirements, economic overhead, and computational efficiency.

3. Related Work

In the past, many studies have published research related to PQC and blockchain, ranging from quantum-resilient consensus mechanisms for secure communication in distributed environments [33, 34, 35]. Some studies evaluated PQC algorithms but focused solely on their mathematical properties or theoretical aspects [36, 37, 38]. Nevertheless, only some studies explored the challenges of integrating PQC with blockchain. Specifically, there has been no research on the performance assessment related to overheads, economic costs and storage requirements. Previous research on hybrid signature mechanisms has been based on theoretical aspects instead of empirical evaluation. For instance, Kiktenko et al. explored quantum threats related to blockchain protocols and proposed quantum-resistant design strategies, but did not provide systematic benchmarking or cost analysis [39]. Similarly, Bindel et al. examined the applicability of PQC algorithms in vehicular communication systems and discussed implications for blockchain-based authentication. Their evaluation encompassed a narrow set of algorithms. However, they did not consider hybrid constructions or on-chain integration [40]. A recent study introduced hybrid signature schemes but lacked comprehensive experimentation [41, 42].

Some blockchain systems such as Algorand [43] and Komodo [44] have studied the integration and performance of post-quantum cryptographic primitives in blockchain systems. They showed that quantum-resilient blockchains are not only theoretically conceivable, but also practically feasible. Meanwhile, academic papers have analyzed the performance of post-quantum signatures in terms

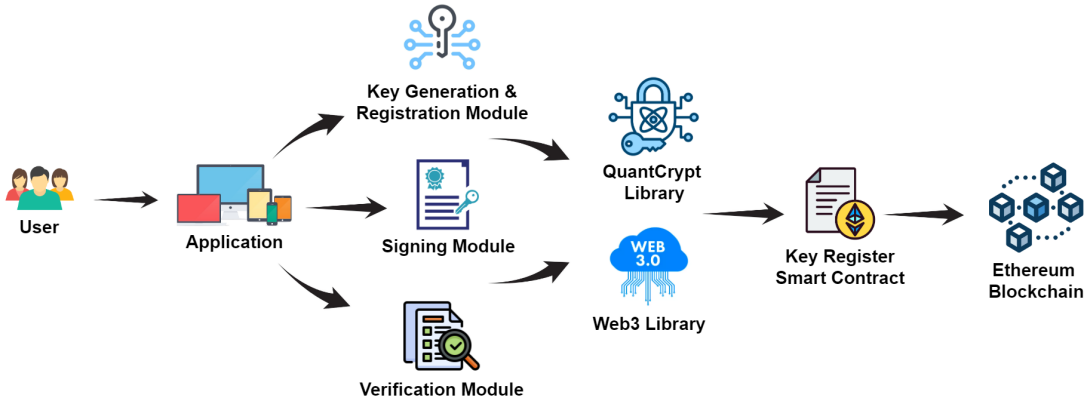


Figure 1: High-level system architecture for hybrid PQC benchmarking in Ethereum.

of computational overhead and transaction latency [45]. Nevertheless, the majority of experiments are carried out on permissive networks or under high-performance environments, limiting insights for low-resource environments. The ongoing research is proceeding towards adapting PQ schemes into blockchain systems, but they focus on individual PQ schemes or signature compressions rather than hybrid classical + PQ signature integration, especially for Ethereum based platforms. While the aforementioned studies dispense valuable benchmarls, only few studies focused on Ethereum-compatible networks or hybrid classical + PQ signatures. Ethereum is moving toward cryptographic robustness and post-quantum-friendly techniques like ZK-STARKs [46, 47, 48] are a part of its future roadmap. In spite of these efforts, there are few practical deployments and performance tests of hybrid signatures on Ethereum especially in low-resource environments [49].

In this research, we introduce a hybrid signature framework for replay resilient Ethereum-like systems that is efficient, and can be verified publicly with the knowledge of the chain. Moreover, we address the gaps that have not been covered by the past studies and extend the research in the domain of PQC in blockchain. We provide a comprehensive benchmarking analysis and evaluate the feasibility of integrating PQC in Ethereum blockchain.

4. Methodology

Our experimental methodology has been designed to characterize, under controlled conditions, the computational and blockchain-level behaviour of a hybrid signature architecture that couples standard ECDSA with NIST-standardized PQ signature schemes. The experimental workflow integrates key generation, hybrid transaction formation, and off-chain verification.

We developed a Python-based framework that orchestrates local blockchain execution, cryptographic routines, and Ethereum transaction handling. A high-level view of the system architecture is depicted in Fig. 1, which illustrates how application-level modules interact with the QuantCrypt library, the Web3 interface, and the KeyRegistry Smart Contract deployed on an Ethereum-compatible blockchain.

4.1. Experimental Framework

The benchmarking framework comprises four principal modules whose interactions mirror the operational sequence of the hybrid signature protocol. The "Key Generation and Registration Module" is responsible for producing both ECDSA and PQC keypairs. PQC key generation is performed via the QuantCrypt library, whereas ECDSA keys follow the `secp256k1` curve used in Ethereum. The module returns the PQC public key alongside the Ethereum address derived from the ECDSA public key. Registration is executed by invoking the KeyRegistry smart contract, which stores PQC public keys on-chain. The corresponding process flow is illustrated in Fig. 2, where keypair derivation, smart-contract invocation, and blockchain-level state changes are shown in sequence.

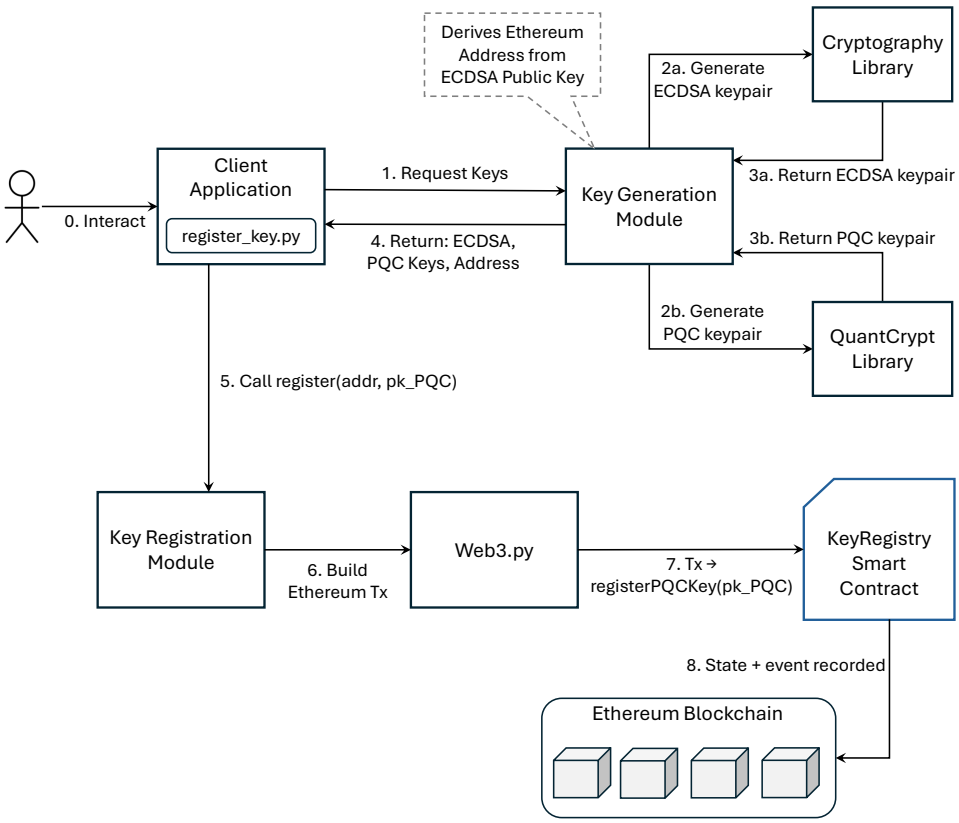


Figure 2: Key Generation and Registration.

The "Signing Module" prepares hybrid transactions by generating a PQC signature on a user-selected message and embedding it into the transaction `calldata`. The Ethereum transaction itself is then constructed and signed with the sender's ECDSA private key through `Web3.py` so that the transaction remains valid under the current Ethereum execution rules. The signing workflow is represented in Fig. 3, which highlights the separation between PQC signing and the ECDSA-based authorization of the transaction envelope. After signing, the "Blockchain Integration Module" (`Web3.py`) submits the signed transaction to a local Ethereum network emulated by Ganache. The ECDSA signature is validated on-chain in the usual manner by the Ethereum nodes, whereas the PQC signature, stored as `calldata`, is recorded in the transaction log. Because no on-chain PQC verification occurs, gas consumption is limited to storage and `calldata` costs.

Finally, the "Verification Module" retrieves PQC signatures and corresponding public keys from on-chain events and smart-contract storage. Verification is executed off-chain using QuantCrypt, which prevents Ethereum-level gas costs while enabling verifiable hybrid transactions. Fig. 4 describes this final stage and the retrieval–verification interplay between blockchain data and local cryptographic routines.

The framework logs all timing data (i.e. key generation, signing, verification) and the size of keys and signatures. Gas consumption is also recorded for each registration and transaction. Each experiment is repeated 30 times to ensure statistical robustness.

4.2. Algorithm Selection

To investigate the performance and cost range associated with PQ signatures, we selected three algorithms representing every family standardized or recommended by NIST. The evaluated algorithms are summarised in Table 1, alongside their security levels and structural characteristics. This selection ensures a representative comparison between compact, fast-verifying lattice-based schemes such as ML-DSA and FALCON, and more conservative hash-based constructions such as SLH-DSA (SPHINCS$^+$).

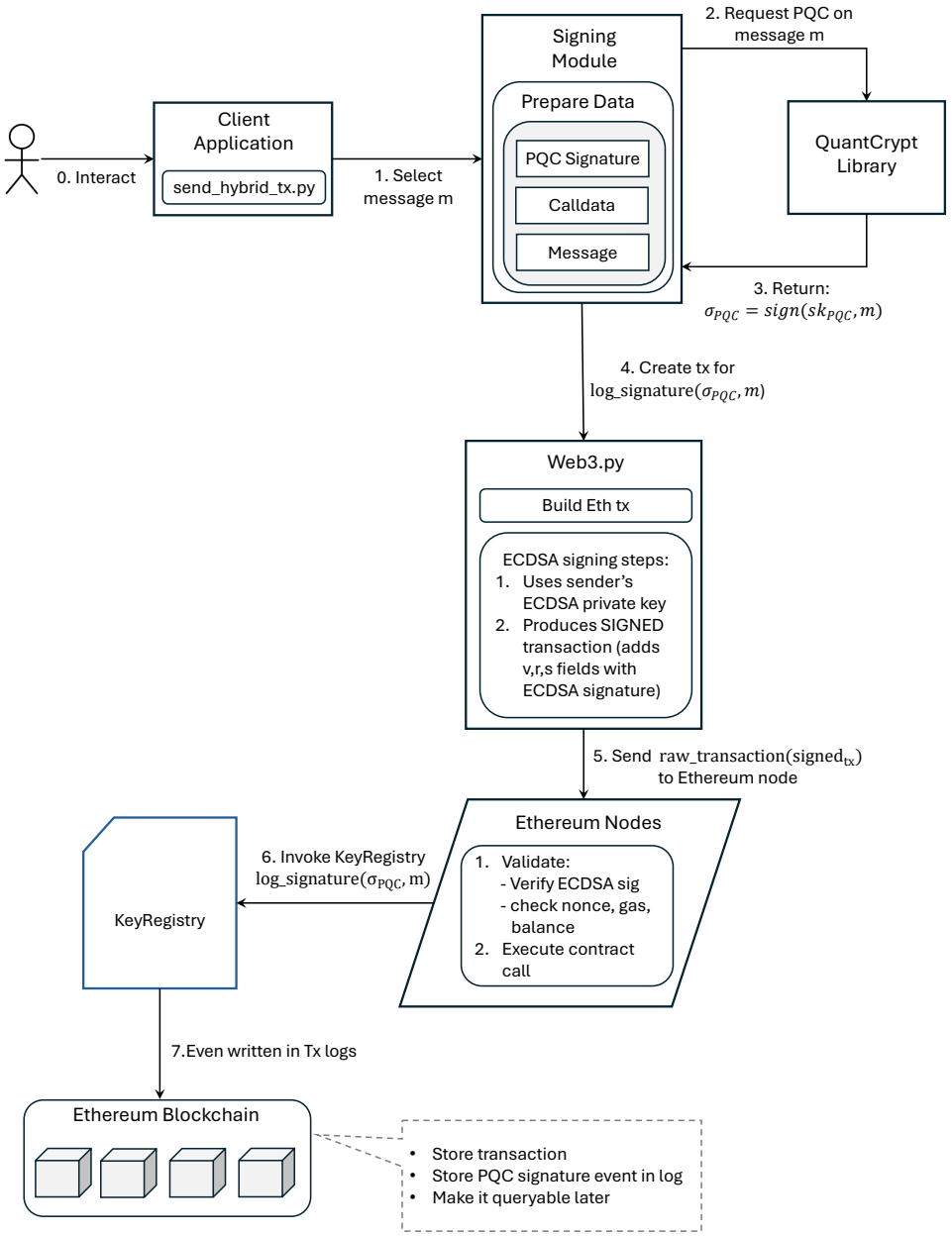


Figure 3: Hybrid Transaction Signing (PQC + ECDSA).

ECDSA (`secp256k1`) serves as the baseline, enabling a direct measurement of the relative computational and storage overhead introduced by PQC.

4.3. Hybrid Signature Architecture

The proposed hybrid architecture binds PQC signatures to conventional Ethereum transactions without altering consensus rules or transaction validity criteria. A conceptual overview is provided in Fig. 1, while Figs. 2, 3 and 4 detail the operational phases. First, each user produces an ECDSA keypair to derive an Ethereum address and generates a PQC keypair to obtain quantum-resistant credentials. The PQC public key is written to the KeyRegistry smart contract, allowing any verifier to retrieve the public key associated with a given Ethereum address. Then, when a user prepares a transaction, the message payload is signed with the PQC private key, and the resulting PQC signature is embedded into the transaction `calldata`. The transaction envelope is subsequently signed with ECDSA to remain fully compliant with Ethereum's current validation rules. The blockchain records both artifacts: the ECDSA

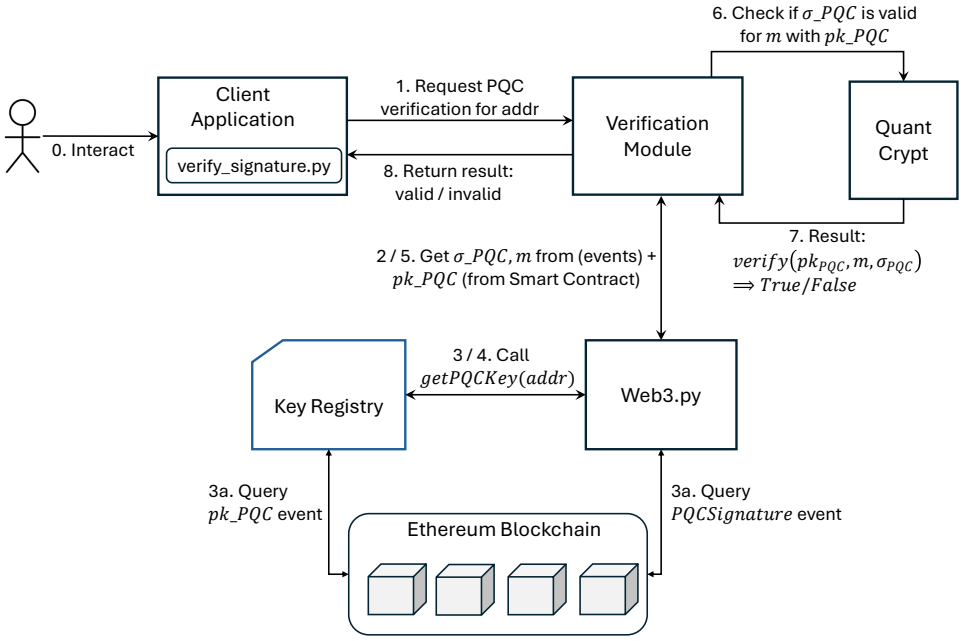


Figure 4: Off-chain PQC Verification.

Table 1
Selected PQC Algorithms.

Algorithm Type	Algorithm Id	Short Description
Lattice-Based	ML-DSA-44	NIST Security Level 2, targeting 128-bit classical security
	ML-DSA-65	NIST Security Level 3, targeting 192-bit classical security
	ML-DSA-87	NIST Security Level 5, targeting 256-bit classical security
Hash-Based	sphincs-shake-256f-simple	NIST Security Level 5; uses SHAKE-256 as internal hash/XOF; SLH-DSA-256 category; fast variant (larger signatures, faster signing/verification); "simple" tweakable-hash SPHINCS$^+$ construction
	sphincs-shake-256s-simple	NIST Security Level 5; uses SHAKE-256 as internal hash/XOF; SLH-DSA-256 category; small variant (smaller signatures, slower signing/verification); "simple" tweakable-hash SPHINCS$^+$ construction
Lattice-Based (NTRU)	FALCON-512	NIST Security Level 1, compact signatures
	FALCON-1024	NIST Security Level 5, higher security
Baseline	ECDSA (secp256k1)	Classical elliptic curve cryptography used in Ethereum

signature authenticates the state transition, whereas the PQC signature provides long-term verifiability against quantum-capable adversaries.

PQC signatures are retrieved exclusively off-chain. This design avoids the prohibitive costs associated with PQC verification inside the EVM, while still enabling external verifiers to authenticate hybrid transactions. To facilitate off-chain verification and potential on-chain verification in the event of protocol changes, PQC public keys are kept in a smart contract mapping. The key benefit of the proposed architecture is its compatibility with the Ethereum blockchain while offering quantum resistance. The architecture thus anticipates future protocol evolutions without imposing premature computational burdens on the blockchain.

Table 2
Evaluation Metrics

Metric Type	Metric Name
Computational Performance	Key generation time (ms) Signing time (ms) Verification time (ms)
Storage Overhead	Public key size (bytes) Private key size (bytes) Signature size (bytes)
Blockchain Costs	Key registration gas cost Transaction gas cost (including PQC signature) Total gas overhead compared to normal transactions

4.4. Evaluation Metrics

Performance analysis considers computational behaviour, storage implications, and gas-level costs, as shown in Table 2.

4.5. Experimental Environment

All experiments were executed on a controlled testbed consisting of a Windows 10 64-bit workstation equipped with a 10-core Intel processor (4C+6G at 2.40 GHz) and 8 GB Random-Access Memory (RAM). The Ethereum execution environment was provided by Ganache [50, 51], enabling deterministic and reproducible transaction execution. PQC operations were performed using the QuantCrypt python implementation [52], and all blockchain interactions (including transaction construction, signing, and broadcast) were managed through Web3.py. Smart-contract functionality was implemented in Solidity 0.8.0 [53] via a custom KeyRegistry smart contract. Running experiments in a local environment eliminates arbitrary network variables and isolates the algorithmic behaviour of PQC schemes. This ensures that the measured effects derives from cryptographic and storage characteristics rather than transient blockchain conditions.

5. Results

This section provides a comprehensive overview of the results obtain from the experiments. First, we present the results of key generation performance. After that, we present the results of signing performance. Next, we present the results of verification performance. Subsequently, we present the results of public key sizes, signature sizes, and private key sizes. Finally, we present the results of key registration costs and transaction costs. Table 3 summarizes the principal performance metrics obtained from the experiments.

5.1. Key Generation Performance

ECDSA keypair generation completes in 4.19 ms on average. Among the PQC schemes, ML-DSA exhibits the lowest key generation times (8.78–16.52 ms), with ML-DSA-44, ML-DSA-65, and ML-DSA-87 at (16.52 ms, (8.78 ms, and 12.39 ms). Similarly, FALCON-512 and FALCON-1024 require 27.94 ms and 63.41 ms, respectively, reflecting the cost of their NTRU-based structure. SLH-DSA (SPHINCS$^+$) shows the widest spread, with the fast variant at 17.26 ms and the small variant at 133.60 ms. This data confirm that lattice-based schemes generally outperform hash-based constructions in key generation. Fig. 5 reports the aggregate comparison, while Fig. 6a contrasts PQC averages with ECDSA. Moreover, to assess scalability, ML-DSA-65 and FALCON-512 were evaluated across batch sizes from 2^1 to 2^{12}. Throughput and latency trends remain stable and scale predictably, as shown in Fig. 6b.

Table 3
Performance Comparison of PQC Signature Algorithms.

Algorithm	Keygen (ms)	Sign (ms)	Verify (ms)	PubKey (B)	Sig (B)	Gas (K)
ECDSA	4.19	3.08	2.14	65	71	N/A
ML-DSA-44	16.52	6.14	3.37	1312	2420	94.4
ML-DSA-65	8.78	8.04	4.49	1952	3309	118.3
ML-DSA-87	12.39	7.76	3.88	2592	4627	153.1
sphincs-shake-256s-simple	133.60	1675.18	5.81	64	29792	828.7
sphincs-shake-256f-simple	17.26	164.31	8.29	64	49856	1374.3
FALCON-512	27.94	6.34	2.82	897	653	47.8
FALCON-1024	63.41	6.99	3.12	1793	1274	64.1

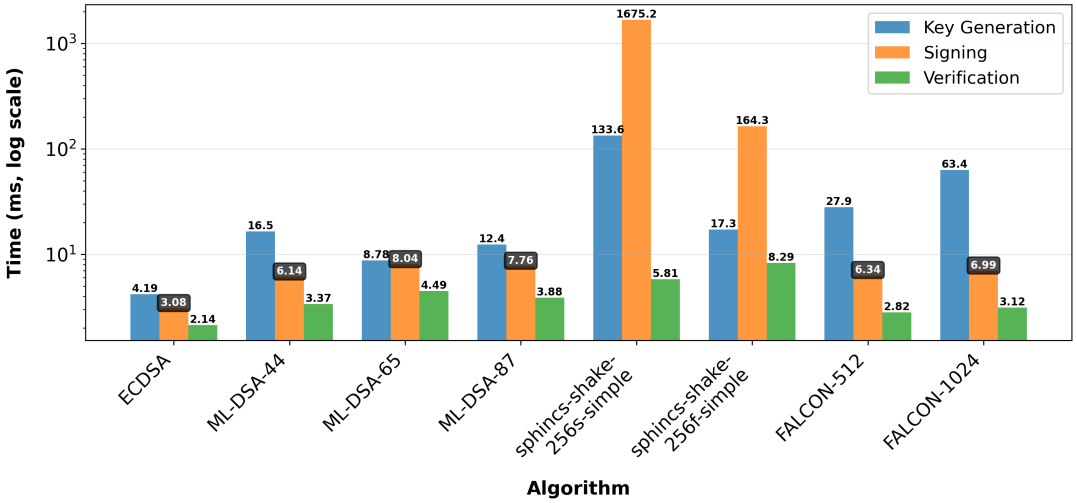


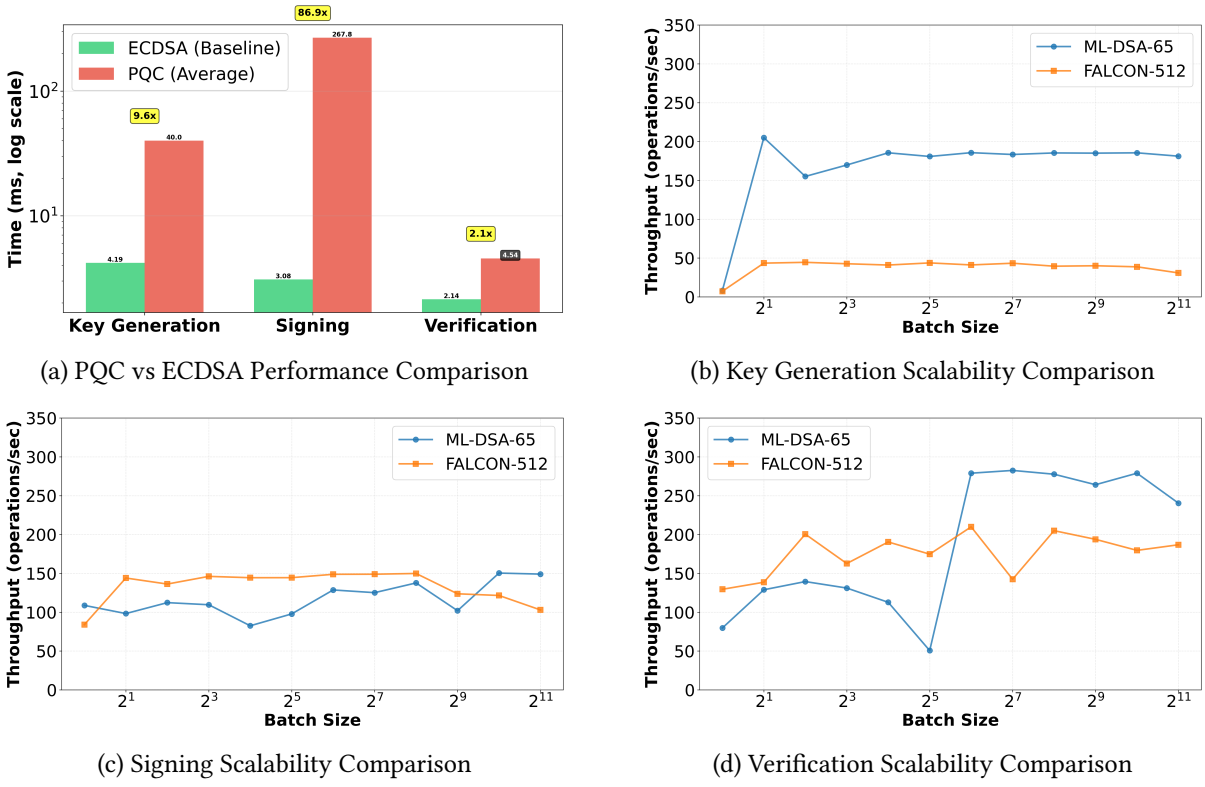
Figure 5: Overall Performance Comparison.

5.2. Signing Performance

ECDSA signing requires 3.08 ms on average. PQC schemes introduce higher but still bounded costs: ML-DSA signs in 6.14–8.04 ms across all parameter sets with ML-DSA-44 having 6.14 ms, ML-DSA-65 having 8.04 ms, and ML-DSA-87 having 7.76 ms. On the other hand, FALCON-512 and FALCON-1024 require 6.34 ms and 6.99 ms. While, SLH-DSA incurs substantially higher costs, with 2164.31 ms for the fast variant and 1675.18 ms for the small variant. For lattice-based schemes, the measured overhead (approximately 2–3× relative to ECDSA) remains feasible for most blockchain workloads, whereas SLH-DSA introduces two to three orders of magnitude overhead. Signing scalability for ML-DSA-65 and FALCON-512 appears in Fig. 6c.

5.3. Verification Performance

ECDSA verification averages 2.14 ms, further proving the efficiency of traditional methods. PQC verification reveals some intriguing trends: ML-DSA algorithms verifies in 3.37–4.49 ms with ML-DSA-44 having 3.37 ms, ML-DSA-65 having 4.49 ms, and ML-DSA-87 having 3.88 ms. While, FALCON having 2.82–3.12 ms, and SLH-DSA having 5.81–8.29 ms. Since verification may happen more often than signing, verification speed is especially crucial for blockchain systems. These results show that lattice-based PQC schemes remain efficient enough for verification-heavy settings, with a 1.3–2.1× increase over ECDSA. SLH-DSA adds additional overhead but remains below its signing cost by a wide margin. Verification scalability for ML-DSA-65 and FALCON-512 is shown in Fig. 6d.



(a) PQC vs ECDSA Performance Comparison

(b) Key Generation Scalability Comparison

(c) Signing Scalability Comparison

(d) Verification Scalability Comparison

Figure 6: Performance comparisons including PQC vs ECDSA, key generation, signing, and verification.

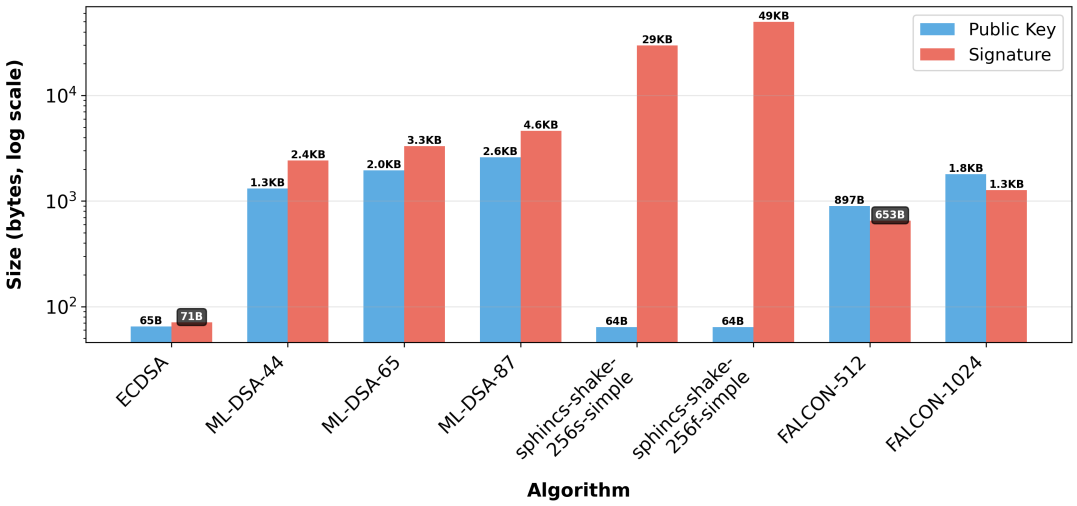


Figure 7: Size Comparison.

5.4. Public Key Sizes

ECDSA public keys have a modest size of 65 B when uncompressed and 33 B when compressed. PQC public keys are significantly larger for lattice-based schemes: 897 B are needed for FALCON-512, 1.793 B for FALCON-1024, 1.312 B for ML-DSA-44, 1.952 B for ML-DSA-65, and 2.592 B for ML-DSA-87. SLH-DSA small and fast variants both use 64 byte public keys. Because these keys must be stored on-chain, their size directly influences registration costs, with lattice-based schemes incurring $10.0\times - 26.8\times$ increases relative to ECDSA. Fig. 7 illustrates the differences.

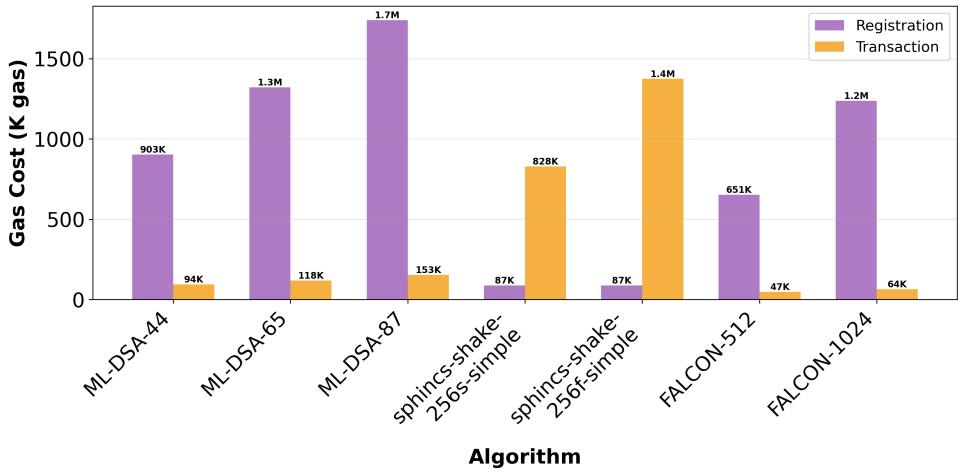


Figure 8: Gas Cost Comparison.

5.5. Signature Sizes

ECDSA signatures are 71 B. PQC signatures are substantially larger: FALCON-512 and FALCON-1024 require 653 and 1274 bytes. While, ML-DSA signatures having 2420 B (ML-DSA-44), 3309 B (ML-DSA-65), and 4627 B (ML-DSA-87). On the other hand, SLH-DSA signatures dominate in size, with 29 792 B for the small variant and 49 856 B for the fast variant. These increases translate directly into `calldata` overhead: ML-DSA and FALCON signatures expand the signature footprint by 34.1 – 65.2× relative to ECDSA, while SLH-DSA increases it by 419.6 – 701.5×, considerably affecting transaction costs and limiting its practical applicability in blockchain systems.

5.6. Private Key Sizes

ECDSA private keys are 32 B. PQC schemes require substantially larger secret keys: 128 B for SLH-DSA 1281 and 2305 bytes for FALCON-512 and FALCON-1024, and 2560–4896 bytes across the ML-DSA parameter sets with ML-DSA-44 having 2560 B, ML-DSA-65 having 4032 B, and ML-DSA-87 having 4896 B. Although private keys are generally kept off-chain, these sizes influence backup strategies, secure hardware requirements, and long-term key management.

5.7. Key Registration Costs

The increase in public key size directly affects on-chain gas consumption. ECDSA key registration typically costs 20–40 K gas. PQC registration costs scale according to key size: ML-DSA-44/65/87 required 903.379, 1321.800, and 1.740.251 gas respectively; FALCON-512 and FALCON-1024 required 651.954 and 1237.791 gas. SLH-DSA exhibits lower registration costs due to its small public keys (87.454 gas for the fast variant and 87.430 gas for the small variant). Fig. 8 presents these comparisons.

5.8. Transaction Costs

A standard Ethereum transaction consumes ~21,000 gas. Adding PQC signatures increases cost proportionally to signature size, resulting in a significant overhead. ML-DSA-44, ML-DSA-65, and ML-DSA-87 signatures add 94 K, 118 K, and 153 K gas. FALCON-512 and FALCON-1024 add 48 K and 64 K gas. SLH-DSA signatures impose the highest overhead, adding 829 K gas (small variant) and 1.374 M gas (fast variant). Depending on the scheme, this results in 2.3 × –65.5× higher transaction costs than standard ECDSA-only transactions.

5.9. Security-Performance Trade-offs

Higher NIST security levels bring predictable increases in runtime and data sizes. ML-DSA-65 and FALCON-1024 offer a balanced compromise between stronger security and manageable computational and economic overhead. ML-DSA-44 and FALCON-512 provide lower-cost alternatives with reduced security margins. ML-DSA-85 and SLH-DSA (parameter set providing NIST security level 5) deliver the strongest protection but introduce the highest costs. Algorithm selection for blockchain systems should carefully weigh longevity requirements, performance constraints, and economic considerations.

5.10. Hybrid Architecture Overhead

In our hybrid signature approach, each transaction carries both an ECDSA and a PQC signature. This introduces additional computation, storage, and `calldata` overhead. However, for lattice-based schemes, this overhead remains tractable, preserving backward compatibility, while enabling incremental migration to quantum-resistant authentication.

6. Discussion and Recommendations

This section provides a comprehensive discussion based on the obtained results and the methodology presented in this research. First, we provide practical deployment implications of hybrid signature approach. Next, we provide PQ based algorithm selection considerations. Subsequently, we provide a discourse on the economic impact of the hybrid signature scheme. Finally, we outline the migration strategy for the hybrid signature scheme. Finally, we provide strategic recommendations for Ethereum.

6.1. Practical Deployment Implications

The benchmarking findings show that the adoption of PQ signatures for hybrid approach is technically feasible and represents an acceptable overhead. The results also indicate that the absolute runtimes for key generation, signing, and verification are higher than for ECDSA. Nevertheless, the measured latencies remain below the thresholds that typically constrain transaction throughput or user-facing responsiveness. In fact, key generation is a rare operation; hence the observed overhead is tolerated even though PQC procedures are more computationally expensive than ECDSA. Moreover, the verification process is the most frequently invoked operation in blockchain infrastructures. The experimental results show that the hybrid approach introduces a modest delay for lattice-based schemes. Hence, the challenge does not lies in the computational costs.

As indicated by the experimental results, the principal operational challenge is based on data footprint that comes from large public keys and signatures. The enlarged public keys and signatures have a direct and predictable influence on storage and `calldata` usage overhead. In systems or environments where transaction execution dominates system behaviour, these costs can be aggregated or bounded through batching, or application-specific compression strategies. Conversely, applications that require sustained high-volume activity or maintain large on-chain registries must account for cumulative storage growth and its long-term effect on network load.

6.2. Algorithm Selection Considerations

The experimental results demonstrate a clear trade-off between cost, performance, and security level. Interestingly, the PQ scheme selection process is strongly related to the target application's performance requirement and security level. Lattice-based constructions like ML-DSA and FALCON always reside in the best part of the trade-off space. They have consistent scaling, stable runtimes, and reasonable `calldata` expansion. Hence, FALCON-512 and ML-DSA44/65 are especially suitable for high-throughput protocols or distributed systems demanding verification-heavy activities.

On the other hand, higher-security variants such as ML-DSA-87 or FALCON-1024 remain attractive for platforms prioritizing long-term cryptographic strength but can materially increase both computational and economic overhead. While hash-based signatures, such as SLH-DSA, remain essential for applications requiring conservative assumptions and long archival lifetimes, they incur significant performance overhead. However, their signature sizes are limiting in bandwidth-constrained contexts and drive high transaction costs. Consequently, they are unlikely to be feasible for general-purpose blockchains, remaining suitable only for specialized, niche, or regulatory domains.

6.3. Economic Impact

According to the gas cost study, because PQC has larger public keys and signatures, the economic overhead to integrate it is not negligible. The cost of registering a key is one-time. But, particularly for lattice-based designs, its magnitude is not negligible. As a result, it could have an impact on how identity-management frameworks and account-abstraction models are designed. Furthermore, in relation to the size of the signature, the transaction-level costs rise linearly. Therefore, a trade-off between application throughput and fee-market conditions is necessary for the economic sustainability of adopting PQC for hybrid approaches.

The experimental results from this work indicate that for most lattice-based methods, the increased fees are still within acceptable ranges for decentralized banking, asset monitoring, or general smart-contract contexts. For hash-based designs, transaction fees rise to levels that have a significant impact on user adoption. Economic factors thus become an explicit part of the algorithm-selection process and ought to be considered in addition to performance and security assurances.

6.4. Migration Strategy

The hybrid signature approach presented in this study provides a practical path into the PQ era while minimizing deployment risks. Moreover, the proposed hybrid architecture is intrinsically backward compatible and preserves the Ethereum semantics. Off-chain PQC verification prevents interference with contract interfaces, execution engines, and consensus clients. The clear separation of roles enables incremental and reversible deployments, enabling PQC artifacts to coexist with conventional signatures until further protocol improvements make native on-chain integration possible.

Our findings highlight the important necessity for explicit governance norms regarding acceptable security levels, algorithm selection, and long-term durability requirements. Standardization organizations and blockchain consortia must coordinate their PQC adoption dates since durability necessitates different levels of assurance and algorithms are not strictly interchangeable. In the absence of this industry-wide consensus, the ecosystem might potentially become extremely fragmented as several participants try to make separate, fragmented transitions to quantum-resilient authentication.

6.5. Strategic Recommendations for Ethereum

Here we provide some strategic recommendation for Ethereum to developers, the academia, and standard bodies.

6.5.1. For Developers

We recommend developers to:

- prioritise PQC requirements during algorithm selection process to balance signature size and performance;
- employ hybrid signature frameworks based on off-chain verification to manage the substantial gas cost implications of large PQC payloads;
- provide fallback compatibility with traditional ECDSA and incorporate phased roadmaps for key management infrastructure into migration procedures.

6.5.2. For Academia

We recommend academia to:

- investigate signature compression and aggregation methods, such as STARK-based wrappers; because payload size is the primary barrier in on-chain gas economics, enhancements in PQ aggregation are necessary prerequisites for native, base-layer standardization;
- explore protocol-level modifications specific to Ethereum's execution environment to reduce the computational load of these algorithms.

6.5.3. For Standard Bodies

We recommend standard bodies to:

- provide detailed migration timelines and interoperability profiles tailored for hybrid PQC architectures in Ethereum networks.

7. Conclusion and Future Research Directions

In this research, we explored the design and integration of PQ signatures into Ethereum. In order to preserve compatibility with Ethereum's validation requirements, we created and assessed a hybrid signature architecture. We benchmarked three NIST-selected PQC algorithms across various security levels, evaluating their impact on computation, storage, and gas-cost of adopting PQ authentication. Hybrid schemes incur longer runtimes and considerably larger keys and signatures than ECDSA. Nevertheless, their absolute performance still remains within acceptable limits for many blockchain applications. The principal overhead arises from `calldata` and storage expansion. Hence, the choice of algorithm must balance long-term security requirements with economic considerations.

Our proposed approach establishes a practical migration path. It allows PQC artifacts to be embedded with minimal changes to existing validation mechanisms. Notably, it does not require fundamental modifications to consensus behaviour. As a result, it allows gradual adoption and allows developers to select schemes that fit their security goals and resource constraints. Our experiments demonstrate that lattice-based designs are the most practical choice for general-purpose deployment. On the other hand, hash-based signatures remain valuable niche applications requiring highly conservative security assumptions.

This study opens several avenues for future research. For example, as the real-world nodes increasingly rely on optimized cryptographic support, hardware-accelerated implementations will need systematic evaluation. Similarly, comprehensive analysis based on protocol-level techniques to mitigate `calldata` expansion such as compression mechanisms or aggregation strategies is also required . Analyzing the security characteristics of hybrid trust models is also essential. Their resilience during long-term cryptographic transitions and their behavior under adversarial circumstances, such as partial key compromise, should be the main focus. Lastly, by capturing the effects of network latency, heterogeneous client behavior, and fee-market dynamics, implementing hybrid signature schemes in live or large-scale test networks would build upon our controlled experiments.

In conclusion, this research has shown that Ethereum can incorporate hybrid PQ authentication without interfering with current operational models. It also draws attention to the crucial trade-offs in design that must be made in order to direct the shift to quantum-resilient infrastructures.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] M. Javaid, A. Haleem, R. Pratap Singh, S. Khan, R. Suman, Blockchain technology applications for Industry 4.0: A literature-based review, Blockchain: Research and Applications 2 (2021) 100027. doi:10.1016/j.bcra.2021.100027.

[2] M. Krichen, M. Ammi, A. Mihoub, M. Almutiq, Blockchain for modern applications: A survey, Sensors 22 (2022) 5274. doi:10.3390/s22145274.

[3] S. Tanwar, Basics of Cryptographic Primitives for Blockchain Development, in: Blockchain Technology: From Theory to Practice, Springer Nature Singapore, 2022, pp. 83–111. doi:10.1007/978-981-19-1488-1_4.

[4] M. Raikwar, S. Wu, Cryptographic Primitives, in: S. Ruj, S. S. Kanhere, M. Conti (Eds.), Blockchains: A Handbook on Fundamentals, Platforms and Applications, volume 105 of *Advances in Information Security*, Springer Cham, Cham (Switzerland), 2023, pp. 25–72. doi:10.1007/978-3-031-32146-7_2, https://link.springer.com/chapter/10.1007/978-3-031-32146-7_2.

[5] T. M. Fernández-Caramès, P. Fraga-Lamas, Towards Post-Quantum Blockchain: A Review on Blockchain Cryptography Resistant to Quantum Computing Attacks, IEEE Access 8 (2020) 21091–21116. doi:10.1109/ACCESS.2020.2968985.

[6] M. El Baraka, E. Siham, Quantum-resistant modifications to ECDSA for blockchain security, Journal of Cyber Security Technology (2025) 1–19. doi:10.1080/23742917.2025.2458320.

[7] A. Joshi, P. Bhalgat, P. Chavan, T. Chaudhari, S. Patil, Guarding Against Quantum Threats: A Survey of Post-Quantum Cryptography Standardization, Techniques, and Current Implementations, in: Applications and Techniques in Information Security (ATIS 2024), Tamil Nadu (India), 2024, pp. 33–46. doi:10.1007/978-981-97-9743-1_3.

[8] D. Ghinea, F. Kaczmarczyck, J. Pullman, J. Cretin, S. Kölbl, R. Misoczki, J.-M. Picod, L. Invernizzi, E. Bursztein, Hybrid Post-quantum Signatures in Hardware Security Keys, in: Applied Cryptography and Network Security Workshops (ACNS 2023), Kyoto (Japan), 2023, pp. 480–499. doi:10.1007/978-3-031-41181-6_26.

[9] 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. doi:10.1038/s41586-022-04623-2.

[10] National Institute of Standards and Technology, Post-Quantum Cryptography, 2017. https://csrc.nist.gov/projects/post-quantum-cryptography.

[11] R. Bavdekar, E. Jayant Chopde, A. Agrawal, A. Bhatia, K. Tiwari, Post Quantum Cryptography: A Review of Techniques, Challenges and Standardizations, in: International Conference on Information Networking (ICOIN), Bangkok (Thailand), 2023, pp. 146–151. doi:10.1109/ICOIN56518.2023.10048976.

[12] National Institute of Standards and Technology, Module-Lattice-Based Key-Encapsulation Mechanism Standard, FIPS 203, 2024. https://csrc.nist.gov/pubs/fips/203/final.

[13] National Institute of Standards and Technology, Module-Lattice-Based Digital Signature Standard, FIPS 204, 2024. https://csrc.nist.gov/pubs/fips/204/final.

[14] National Institute of Standards and Technology, Stateless Hash-Based Digital Signature Standard, FIPS 205, 2024. https://csrc.nist.gov/pubs/fips/205/final.

[15] H. Nguyen, B. Cambou, T. T. Nguyen, A GPU-Accelerated High-Performance Design for CRYSTALS-Dilithium Digital Signature, in: IEEE International Conference on Consumer Electronics (ICCE), Las Vegas (NV, USA), 2025, pp. 1–4. doi:10.1109/ICCE63647.2025.10929968.

[16] M. Raavi, Q. Khan, S. Wuthier, P. Chandramouli, Y. Balytskyi, S.-Y. Chang, Security and performance analyses of post-quantum digital signature algorithms and their TLS and PKI integrations, Cryptography 9 (2025) 38. doi:10.3390/cryptography9020038.

[17] Agrawal, M., Duraisamy, K., Ganesan, K.S., Gupta, S., Kandele, S., Konduru, S.S., Maddipati, H.C., Raghavendra, K., Sahu, R.A. and Saraswat, V., Secure Boot in Post-Quantum Era, in: 24th International Conference on Cryptology (INDOCRYPT), Goa (India), 2023, pp. 223–239. doi:10.1007/978-3-031-56235-8_11.

[18] D. Kim, H. Choi, S. C. Seo, Parallel Implementation of SPHINCS+ with GPUs, IEEE Transactions on Circuits and Systems I: Regular Papers 71 (2024) 2810–2823. doi:10.1109/TCSI.2024.3370802.
[19] B. Appiah, D. Commey, I. Osei, B. Frimpong, G. Assamah, E. Hammond, Secure IoT Firmware Updates Against Supply Chain Attacks, The Journal of Supercomputing 81 (2025) 1–20. doi:10.1007/s11227-025-07126-9.
[20] N. Q. Luc, T. T. Nguyen, D. H. Quach, T. T. Dao, N.-T. Pham, Building Applications and Developing Digital Signature Devices Based on the FALCON Post-Quantum Digital Signature Scheme, Engineering, Technology & Applied Science Research 13 (2023) 10401–10406. doi:10.48084/etasr.5674.
[21] M. A. Khan, M. A. Khan, Securing IoT in the Post-Quantum Era: Implementation, Challenges and Future Directions, IEEE Communications Standards Magazine (2025) 1–7. doi:10.1109/MCOMSTD.2025.3584658.
[22] N.-Q. Luc, T.-T. Nguyen, C.-H. Vu, D.-H. Quach, T.-T. Dao, Secure Messaging Application Development: Based on Post-Quantum Algorithms CSIDH, Falcon, and AES Symmetric Key Cryptosystem, Programming and Computing Software 50 (2024) 322–333. doi:10.1134/S0361768824700130.
[23] Global Quantum Intelligence (GQI), Quantum Computing Report, 2022. https://quantumcomputingreport.com/nist-selects-hqc-as-backup-post-quantum-encryption-algorithm/.
[24] W. Zhang, T. Anand, Ethereum Architecture and Overview, in: Blockchain and Ethereum Smart Contract Solution Development: Dapp Programming with Solidity, Apress, Berkeley, CA, USA, 2022, pp. 209–244. doi:10.1007/978-1-4842-8164-2.
[25] J. Liu, Digital Signature and Hash Algorithms Used in Bitcoin and Ethereum, in: Third International Conference on Machine Learning and Computer Application (ICMLCA 2022), Shenyang (China), 2023, pp. 1302–1321. doi:10.1117/12.2675431.
[26] Ethereum Foundation, Ethereum Roadmap, 2025. https://ethereum.org/roadmap/.
[27] S. Ricci, P. Dobias, L. Malina, J. Hajny, P. Jedlicka, Hybrid keys in practice: Combining classical, quantum and post-quantum cryptography, IEEE Access 12 (2024) 23206–23219. doi:10.1109/ACCESS.2024.3364520.
[28] K. Cherkaoui Dekkaki, I. Tasic, M.-D. Cano, Exploring post-quantum cryptography: Review and directions for the transition process, Technologies 12 (2024) 241. doi:10.3390/technologies12120241.
[29] V. Buterin, Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform, 2014. White Paper, https://ethereum.org/en/whitepaper/.
[30] J. J. Kearney, C. A. Perez-Delgado, Vulnerability of blockchain technologies to quantum attacks, Array 10 (2021) 100065. doi:10.1016/j.array.2021.100065.
[31] Y. Baseri, A. Hafid, Y. Shahsavari, D. Makrakis, H. Khodaiemehr, Blockchain Security Risk Assessment in Quantum Era, Migration Strategies and Proactive Defense, IEEE Communications Surveys & Tutorials (2025). doi:10.1109/COMST.2025.3621113.
[32] N. Dey, M. Ghosh, A. Chakrabarti, Quantum Solutions to Possible Challenges of Blockchain Technology, in: A. Kumar, S. S. Gill, A. Abraham (Eds.), Quantum and Blockchain for Modern Computing Systems: Vision and Advancements: Quantum and Blockchain Technologies: Current Trends and Challenges, Springer International Publishing, 2022, pp. 249–282. doi:10.1007/978-3-031-04613-1_9.
[33] M. S. Peelam, V. Chamola, B. Sikdar, Enhancing Security Using Quantum Blockchain in Consumer IoT Networks, IEEE Transactions on Consumer Electronics 71 (2024) 4819–4837. doi:10.1109/TCE.2024.3512791.
[34] J. Gomes, S. Khan, D. Svetinovic, Fortifying the Blockchain: A Systematic Review and Classification of Post-Quantum Consensus Solutions for Enhanced Security and Resilience, IEEE Access 11 (2023) 74088–74100. doi:10.1109/ACCESS.2023.3296559.
[35] M. J. Thompson, E. R. Carter, B. A. Wallace, S. L. Bennett, C. James, Quantum Threats to Blockchain-Based Financial Platforms, 2021. https://www.researchgate.net/publication/396968817_Quantum_Threats_to_Blockchain-Based_Financial_Platforms.
[36] M. Zohaib, F. S. Altuwaijri, S. Hyrynsalmi, Integrating Quantum Computing and Blockchain:

Building the Foundations of Secure, Efficient 6G Technology, in: 1st ACM International Workshop on Quantum Software Engineering: The Next Evolution (QSE-NE 2024), Porto de Galinhas (Brazil), 2024, pp. 27–34. doi:10.1145/3663531.3664755.

[37] T. Srivastava, B. Bhushan, S. Bhatt, A. K. M. B. Haque, Integration of Quantum Computing and Blockchain Technology: A Cryptographic Perspective, in: R. Kumar, R. Sharma, P. K. Pattnaik (Eds.), Multimedia Technologies in the Internet of Things Environment, Volume 3, Springer Singapore, 2022, pp. 197–228. doi:10.1007/978-981-19-0924-5_12.

[38] Z. Yang, T. Salman, R. Jain, R. D. Pietro, Decentralization Using Quantum Blockchain: A Theoretical Analysis, IEEE Transactions on Quantum Engineering 3 (2022) 1–16. doi:10.1109/TQE.2022.3207111.

[39] E. O. Kiktenko, N. O. Pozhar, M. N. Anufriev, A. S. Trushechkin, R. R. Yunusov, Y. V. Kurochkin, A. I. Lvovsky, A. K. Fedorov, Quantum-secured blockchain, Quantum Science and Technology 3 (2018). doi:10.1088/2058-9565/aabc6b.

[40] G. Twardokus, N. Bindel, H. Rahbari, S. McCarthy, When Cryptography Needs a Hand: Practical Post-Quantum Authentication for V2V Communications, in: 31st Annual Network and Distributed System Security Symposium (NDSS), San Diego (CA, USA), 2024, pp. 1–20. doi:10.14722/ndss.2024.24267.

[41] H.-Y. Kwon, I. Bajuna, M.-K. Lee, Compact Hybrid Signature for Secure Transition to Post-Quantum Era, IEEE Access 12 (2024) 39417–39429. doi:10.1109/ACCESS.2024.3374645.

[42] R. Amos, M. Georgiou, A. Kiayias, M. Zhandry, One-shot signatures and applications to hybrid quantum/classical authentication, in: 52nd Annual ACM SIGACT Symposium on Theory of Computing (STOC 2020), Chicago (IL, USA), 2020, pp. 255–268. doi:10.1145/3357713.3384304.

[43] Algorand Foundation, Post-Quantum Security on Algorand, 2023. https://algorand.co/technology/post-quantum.

[44] Komodo Platform, Dilithium: Quantum-Secure Blockchain Signatures, 2022. https://komodoplatform.com/en/blog/dilithium-quantum-secure-blockchain/.

[45] Brazilian Computer Society (SBC), Post-Quantum Cryptography in Blockchain Systems, 2021. https://sol.sbc.org.br/index.php/sbseg/article/download/19229/19058/.

[46] Ethereum Foundation, Ethereum Roadmap: Future-Proofing, 2023. https://ethereum.org/roadmap/future-proofing/.

[47] StarkWare Industries, STARK Proofs: Scalable and Transparent Arguments of Knowledge, 2022. https://starkware.co/stark/.

[48] StarkNet Foundation, What Is StarkNet?, 2022. https://www.starknet.io/what-is-starknet/.

[49] Technology Innovation Institute (TII), PQSort: Post-Quantum Cryptography Benchmarking Platform, 2023. https://pqsort.tii.ae/.

[50] Truffle Suite, Ganache: One Click Blockchain, 2021. https://archive.trufflesuite.com/ganache/.

[51] Truffle Suite, Ganache – GitHub Repository, 2021. https://github.com/ConsenSys-archive/ganache.

[52] Mattias Aabmets, QuantCrypt: Cross-platform Python library for Post-Quantum Cryptography using precompiled PQClean binaries, 2023. https://pypi.org/project/quantcrypt/.

[53] Ethereum Foundation, Solidity v0.8.0, 2020. https://docs.soliditylang.org/en/v0.8.0/.