Crypto-Agile Ethereum Transactions: A Hybrid Post-Quantum Architecture for Besu

J.B.S. Dantas[1], E.M. Silveira[1], R. Fontes[2], A. Cruz[2], A.F.P. dos Santos[1,4], J.F. Bragança[1], D.H. Moraes[1], R. Fava[2], L. Rosendo[2], J. Petry[2], M.C. de Oliveira[3], F.F. Fanchini[3], L.G.E. Arruda[3], D.G. de Oliveira[1]

[1] Venturus Centro de Inovação Tecnológica, Campinas, SP – Brazil
[2]Universidade Federal do Rio Grande do Norte (UFRN), Natal, RN – Brazil
[3]QuaTi – Quantum Technology & Information, São Paulo, SP – Brazil
[4]Instituto Militar de Engenharia (IME), Rio de Janeiro, RJ – Brazil

joao.dantas@venturus.org.br

Abstract. *Quantum computing represents a critical threat to classical ECDSA-based blockchains. This paper introduces a crypto-agile framework natively integrated into Besu, proposing novel transaction types that support hybrid post-quantum signatures (e.g., ECDSA-MLDSA44 using the Strong Nesting combiner) while addressing the lack of public key recovery. Evaluated under QBFT consensus, the results show that while local transaction pool ingestion remains performant, the cumulative overhead of validating large cryptographic payloads significantly delays block finalization and degrades global network throughput. This demonstrates that the primary bottleneck for post-quantum migration lies within the consensus layer rather than local transaction routing, offering key insights for future ledger optimization.*

1. Introduction

Modern distributed architectures and communication networks heavily rely on public-key cryptographic protocols, such as RSA and Diffie-Hellman, to safeguard data exchange and user privacy [Imam et al. 2021]. The core security properties of these systems depend on the fundamental assumption that specialized mathematical operations, primarily integer factorization, order finding, and the discrete logarithm problem, are computationally intractable when processed via traditional mechanisms within humanly viable timeframes. Under classical computation paradigms, the most advanced algorithms capable of solving these underlying mathematical puzzles exhibit super-polynomial or exponential execution complexities, requiring a vast amount of time relative to the size of the input data [Dell et al. 2014]. Because even symmetric cryptosystems frequently leverage public-key infrastructure to coordinate secure initial key exchanges or establish shared secret parameters, these classical hardness assumptions serve as the structural foundation for all of today's digital security systems.

However, this foundational baseline has been deeply challenged by the maturation of quantum computation architectures. Theoretical benchmarks show that quantum-mechanical hardware can effectively execute the Quantum Fourier Transform (QFT) on input vectors, achieving an exponential speedup compared with the performance limits of classical systems [Nielsen and Chuang 2010]. Capitalizing on this fundamental advantage, Shor [Shor 1999] designed a specialized quantum algorithm capable of computing

prime factorizations in polynomial time. This architectural paradigm shift was subsequently generalized to solve discrete logarithms and order-finding tasks with equivalent polynomial-time efficiency, effectively proving that a sufficiently mature quantum adversary can compromise the mathematical foundations of traditional asymmetric cryptography [Nielsen and Chuang 2010].

To counteract this systemic vulnerability, the field of Post-Quantum Cryptography (PQC) has emerged, focusing on the formulation and deployment of new cryptographic algorithms designed to resist attacks by both classical and cryptographically relevant quantum hardware [Fernandez-Carames and Fraga-Lamas 2020]. These post-quantum schemes are built on complex multi-dimensional mathematical structures such as structured lattices, error-correcting codes, multivariate quadratic equations, or cryptographic hash trees, which remain inherently robust against quantum-accelerated resolution techniques [Fernandez-Carames and Fraga-Lamas 2020]. Driven by the critical necessity for standardized alternatives, the National Institute of Standards and Technology (NIST) initiated an exhaustive evaluation process to select and implement secure post-quantum primitives [National Institute of Standards and Technology].

This global initiative culminated in the publication of several official standards for digital signatures, notably FIPS 204, which regulates the Module-Lattice-Based Digital Signature Algorithm (ML-DSA, originally known as Dilithium), and FIPS 205, which formalizes the Stateless Hash-Based Digital Signature Algorithm (SLH-DSA, formerly designated as SPHINCS+) [National Institute of Standards and Technology (NIST) 2024]. Furthermore, the Falcon lattice-based scheme [Fouque et al. 2017] has been designated for imminent release under the FIPS 206 standard as the Fast Fourier Transform over NTRU-Lattice-Based Digital Signature Algorithm (FN-DSA).

The structural obsolescence of classical asymmetric primitives introduces profound operational risks for decentralized ledger networks, which utilize digital signature mechanisms to enforce transaction validation, structural immutability, and strict non-repudiation invariants. Popular blockchain environments commonly rely on the Elliptic Curve Digital Signature Algorithm (ECDSA), executing cryptographic computations over curves such as secp256k1 or secp256r1 [Fernandez-Carames and Fraga-Lamas 2020]. Because public-key values are recorded directly on distributed ledgers during standard transaction propagation, quantum adversaries could extract the corresponding private keys, compromising account integrity. Consequently, upgrading production-grade distributed architectures to support post-quantum signature schemes represents an urgent requirement to guarantee the long-term historical persistence and resilience of blockchain frameworks.

1.1. Motivation and Contribution

The motivation for this work arises from the risk that quantum computing technologies pose to the cryptographic foundations and, consequently, to the integrity and authenticity of blockchain-based infrastructures. Currently, major platforms such as Bitcoin and Ethereum rely on the ECDSA over `secp256k1` curve, which is inherently vulnerable to quantum attacks capable of deriving private keys from publicly available ledger data. Given the immutable nature of distributed ledgers, transitioning to quantum-resistant digital signatures is essential to ensure long-term security and resilience. While

the NIST has standardized several post-quantum algorithms (such as ML-DSA), their practical adoption faces significant performance challenges, including larger key sizes, increased signature payloads, and substantial computational overhead compared to classical schemes.

To address these challenges, this paper introduces a crypto-agile framework integrated into the Besu Ethereum[1] client. The main contributions of this work are threefold:

- **Architecture Design and Core Prototype:** we design and implement a standard-agnostic, backward-compatible framework within Besu. By introducing a family of `Flexible` transaction types, the architecture decouples transactions from rigid classical constraints, mitigating the lack of Public Key Recovery (PKR) in post-quantum primitives through explicit dual-layer verification paths;
- **Dynamic Cryptographic Agility Control:** an on-chain/node-level governance mechanism driven by configuration flags (`accounts-supported-dsas`). This allows network operators to dynamically authorize, deploy, or deprecate digital signature algorithms (DSAs) without requiring disruptive client-side code modifications or rewriting historical blocks;
- **Public testbed:** a public testbed environment designed to orchestrate post-quantum benchmarks under the QBFT (Quorum Byzantine Fault Tolerant) consensus mechanism, enabling comprehensive replication and future distributed ledger scalability research.

The remainder of this paper is organized as follows: Section 2 provides the theoretical background on quantum threats, post-quantum cryptography families, and digital signature vulnerabilities in blockchain systems. Section 3 reviews relevant works in the domain of quantum-resistant ledgers. Section 4 details our proposed crypto-agile architecture, including the hybrid account structures, the new flexible transaction encodings, and the step-by-step validation flow. Section 5 presents the experimental validation, detailing the benchmark environment and discussing the performance results. Finally, Section 6 concludes the paper with final considerations and outlines prospective paths for future work.

2. Background

This section establishes the theoretical foundation for understanding the transition from classical cryptographic security to quantum-resistant infrastructure. A comprehensive understanding of these concepts is essential to analyze how the maturation of quantum computing fundamentally alters the security assumptions currently accepted in distributed systems.

2.1. Classical Cryptography and Quantum Threats

Public-key cryptography constitutes the foundation of secure communication in modern distributed systems. Widely adopted schemes, such as RSA and Elliptic Curve Cryptography (ECC), rely on the computational hardness of problems including integer factorization and the discrete logarithm problem. These assumptions provide essential security properties, including confidentiality, authentication, and non-repudiation, which are extensively used in secure communication protocols and digital signature schemes. Currently, the most efficient classical algorithms for solving

[1] https://besu.hyperledger.org/

these problems exhibit super-polynomial or exponential complexity, typically around $O(n^2 \log n \log\log n)$ [Shor 1999].

However, quantum computing introduces a paradigm shift in computational capabilities, leveraging principles such as superposition and entanglement to achieve significant speedups for specific problems. In the context of asymmetric cryptography, the advantage of quantum computation lies in its ability to efficiently perform the QFT on vectors of size $N = 2^n$ in $O(n^2)$ operations [Nielsen and Chuang 2010]. In particular, Shor's algorithm [Shor 1999] employs the QFT to efficiently solve integer factorization and discrete logarithm problems in polynomial time with respect to the bit length n, thereby effectively compromising the security of widely used public-key cryptosystems.

As a consequence, cryptographic primitives that are currently considered secure may become obsolete in the presence of a Cryptographically Relevant Quantum Computer (CRQC). This creates a critical long-term security risk, often referred to as "harvest now, decrypt later" (HNDL), especially given the immutable nature of data in systems like blockchain [Fernandez-Carames and Fraga-Lamas 2020]. Therefore, transitioning to post-quantum digital signatures is essential to guarantee the long-term security and resilience of modern digital infrastructures.

2.2. Post-Quantum Cryptography

PQC focuses on the design and implementation of cryptographic schemes that remain secure against both classical and quantum adversaries. Unlike current public-key standards, these schemes are underpinned by mathematical problems that are widely believed to be resistant to efficient solutions, even in the presence of a CRQC. The main families of PQC include lattice-based, code-based, multivariate, hash-based, and isogeny-based cryptography. Each of these families offers distinct trade-offs regarding security levels, computational performance, and key sizes, each suited to an application scenario such as high-performance servers or resource-constrained IoT devices.

The global demand for quantum-resistant solutions has motivated rigorous initiatives led by the NIST to evaluate and standardize secure post-quantum algorithms. This ongoing standardization process has already resulted in the approval of several digital signature schemes. Specifically, FIPS 204 defines the Module-Lattice-Based Digital Signature Algorithm (ML-DSA), previously known as Dilithium [National Institute of Standards and Technology]. Additionally, FIPS 205 specifies the Stateless Hash-Based Digital Signature Algorithm (SLH-DSA), formerly referred to as SPHINCS+. Another prominent candidate is the Falcon algorithm [Fouque et al. 2017], which is expected to be published as FIPS 206 under the name Fast Fourier Transform over NTRU-Lattice-Based Digital Signature Algorithm (FN-DSA). These algorithms are slated to replace traditional schemes like ECDSA to maintain the integrity and trustworthiness of critical infrastructures, particularly in decentralized environments.

However, the transition to these new algorithm families is not immediate and involves significant trust and implementation challenges. In this scenario, hybrid cryptographic schemes [Fedorov 2023, Zeng et al. 2024] emerge as a fundamental strategy to ensure security during the migration period. A hybrid scheme combines the security of a time-tested classical algorithm, such as ECDSA, with the quantum resistance of new primitives like ML-DSA or SLH-DSA. This approach ensures that the system remains

protected even if a vulnerability is discovered in the new post-quantum standards or if an attacker utilizes advanced classical computation. Consequently, it maintains backward compatibility and the robustness of the digital infrastructure during the maturation of PQC.

2.3. Digital Signatures in Blockchain Systems

Blockchain systems rely heavily on digital signatures to ensure fundamental security properties, including transaction authenticity, integrity, and non-repudiation. On major platforms such as Bitcoin and Ethereum, users authenticate transactions and authorize transfers using elliptic-curve-based signature schemes, specifically the ECDSA. However, these widely deployed schemes are inherently vulnerable to quantum attacks. A sufficiently powerful quantum computer, leveraging algorithms such as Shor's, could efficiently derive private keys from publicly available information on the ledger.

This vulnerability introduces a critical long-term security risk: HNDL [Mosca 2018]. Since blockchain data is immutable and stored across distributed ledgers, any information recorded today remains exposed to future decryption once large-scale quantum computers become viable. Consequently, the persistence of sensitive data in these systems necessitates an urgent transition to post-quantum digital signature schemes. Adopting quantum-resistant primitives is essential to guarantee the long-term resilience, security, and continued trustworthiness of blockchain infrastructures against both current and future cryptographic threats [Fernandez-Carames and Fraga-Lamas 2020].

In this context, hybrid cryptographic schemes also provide a viable migration path, combining the established reliability of ECDSA with the quantum resistance of PQC primitives [Ghinea et al. 2023, Nouma and Yavuz 2024]. By using a dual-layer signature, blockchain infrastructure can maintain backward compatibility while ensuring transactions remain secure even if one of the underlying algorithms is compromised. Adopting these quantum-resistant hybrid primitives is essential to guarantee the long-term resilience, security, and continued trustworthiness of blockchain infrastructures against both current and future cryptographic threats

3. Related Works

Multiple real-world systems have started integrating PQC primitives to proactively address emerging threats posed by quantum computing in the blockchain domain. For example, the Quantum Resistant Ledger (QRL) project [Waterland 2016] has implemented XMSS hash-based signatures, making it one of the earliest public ledgers secured with post-quantum cryptography; the Ethereum Foundation's roadmap includes EIP-4844 [Buterin et al. 2022], which establishes a foundation for future post-quantum integration; and the Blockchain-based Service Network (BSN) has published pilot deployment reports showcasing the use of TLS channels secured with post-quantum mechanisms based on CRYSTALS-Kyber [State Information Center of China 2023], highlighting its readiness for large-scale industrial adoption.

The rise of quantum computing has intensified research into blockchain security, particularly in areas such as quantum-resistant cryptography, secure key exchange, and consensus mechanisms. Existing studies have independently explored post-quantum cryptosystems, quantum key distribution (QKD) protocols, and enhancements to

blockchain architectures. Several architectures have been proposed to incorporate quantum technologies into blockchain systems. These approaches include quantum tokenization, quantum consensus mechanisms, and entanglement-based blockchain designs, all aiming to strengthen decentralization and immutability while providing resistance against quantum attacks.

[Gao et al. 2020] introduced a quantum blockchain model focused on improving security and efficiency, particularly in mitigating quantum threats, although scalability remains a key challenge for future investigation. [Edwards et al. 2020] analyzed the evolution of quantum blockchain systems, emphasizing aspects such as sustainability, scalability, and efficiency, while also identifying open issues and research gaps. Focusing on decentralized identity, [Yang et al. 2023] explored the use of quantum blockchain for identity verification, highlighting current limitations and proposing directions for building quantum-resistant infrastructures. In a similar vein, [Nilesh and Panigrahi 2022] examined quantum security challenges and proposed a conceptual framework for quantum blockchain, outlining future research opportunities, especially in the development of quantum tokens.

[Zohaib et al. 2024] extended the application of quantum blockchain to smart city contexts by proposing a Quantum-Blockchain-6G architecture, addressing scalability, integration, and cryptographic risks. Additionally, [Iovane 2021] proposed a quantum-based negotiation mechanism for blockchain validation, tackling issues related to randomness, fairness, and secure key distribution. Finally, [Abulkasim et al. 2021] developed a quantum-based sealed-bid auction protocol that ensures security, anonymity, and comprehensive functionality.

Some approaches further combine QKD with PQC to create hybrid frameworks, thereby enhancing the security of blockchain data transmission. For example, [Dhar et al. 2024] investigated the use of blockchain and quantum cryptography to improve multimedia security in IoT environments, identifying computational constraints as a key limitation and highlighting opportunities for advancing security mechanisms in future work. [Gupta et al. 2023] examined the vulnerability of traditional blockchain systems to quantum attacks and proposed a quantum-secure electronic voting scheme, while also outlining promising research directions.

[Radanliev 2024] explored the role of artificial intelligence in strengthening quantum cryptographic systems, discussing its potential integration into broader digital security frameworks and addressing associated challenges. In the context of smart cities, [Chen et al. 2021] proposed a post-quantum Proof-of-Work (PoW) mechanism to mitigate quantum threats, while improving both security and transaction throughput. [Pedone et al. 2021] addressed emerging quantum-related challenges by introducing a QKD software stack and simulation environment designed for secure integration into distributed systems. Also, [Garcia et al. 2024] proposed hybrid QKD-PQC architectures aimed at achieving long-term, quantum-resistant security for protocols such as TLS, acknowledging, however, the performance trade-offs associated with these solutions. Finally, [de Haro Moraes et al. 2024] combines post-quantum signatures and Trusted Execution Environment (TEE) to safeguard the confidentiality of transaction contents in a Central Bank Digital Currency (CBDC) infrastructure based on Distributed Ledger Technology, showing that lattice-based schemes present a favorable balance between compu-

tational cost and communication footprint in the evaluated CBDC scenario.

Despite the significant advancements in quantum-resistant architectures and theoretical blockchain models, a notable gap remains in executing a seamless, backward-compatible transition within production-grade, EVM-equivalent distributed ledgers. Most existing works either propose clean-slate alternative protocols or focus on high-level conceptual frameworks that overlook the engineering complexities of real-world client implementations. This work directly addresses this gap by introducing a concrete, crypto-agile architecture natively integrated into the Besu Ethereum client.

Moreover, unlike prior solutions, our proposal decouples transaction and authorization structures from rigid classical standards through a family of novel `Flexible` transaction types. This approach not only resolves practical constraints such as the lack of PKR in modern post-quantum primitives via explicit dual-layer verification paths, but also establishes an immediate, node-level governance mechanism driven by configuration flags. Consequently, our framework enables network operators to progressively rollout quantum-safe security without disrupting current blockchain operations or altering historical data.

The choice of Besu as the foundational platform is highly strategic; as an enterprise-grade, open-source Ethereum client governed by the Linux Foundation, Besu is widely recognized as a premier infrastructure for private, permissioned networks and consortium blockchains globally. It serves as the primary technical backbone for major institutional distributed ledger deployments, including Central Bank Digital Currency (CBDC) pilots such as the Brazilian Drex initiative and cross-border regional networks like LACChain[2]. Consequently, modifying and benchmarking Besu ensures that the findings and the proposed hybrid transaction architecture of this study are immediately relevant to real-world, highly regulated production environments.

4. Proposal

The proposal for this work consists of implementing and evaluating a hybrid cryptographic scheme that combines classical and post-quantum algorithms to provide security against both current and future threats. Typically, a hybrid scheme performs multiple cryptographic operations, such as signing or key exchange, using different primitives to ensure that the system remains secure even if one component is compromised.

Figure 1 illustrates the integration of hybrid signature schemes within a blockchain environment. The diagram details the structure of Externally Owned Accounts (EOAs), the support for multiple Digital Signature Algorithms (DSAs), and the newly introduced transaction types. The core components of this integration are described below:

- **Supported DSAs**: the system provides support for classical standards such as ECDSA/SECP256K1 and ECDSA/SECP256R1, as well as the proposed hybrid scheme, ECDSA/SECP256K1-MLDSA44;
- **Hybrid Account Structure (EOA2)**: unlike a conventional account (EOA1) that uses a single public key derived from the signature ($pk = PKR(message, signature)$), the hybrid account supports a composite public key. This key is formed by the concatenation of the classical and post-quantum keys ($firstPk \parallel secondPk$);

[2]github.com/lacchain

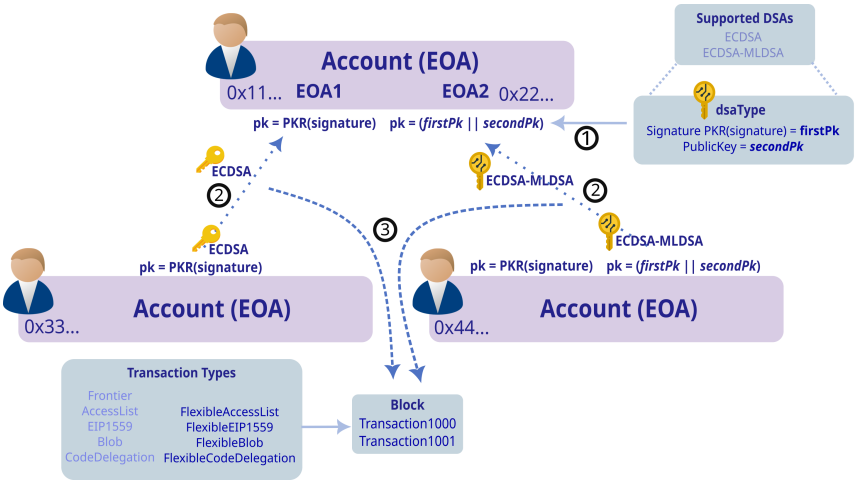


Figure 1. Hybrid Account Structure and Transaction Types

- **Transaction Types**: the architecture introduces new "flexible" transaction types (such as FlexibleAccessList, FlexibleEIP1559, FlexibleBlob, and FlexibleCodeDelegation) designed to accommodate the additional metadata required by post-quantum cryptographic schemes;

A core feature of the proposed architecture design is its ability to maintain full compatibility with traditional cryptographic systems while simultaneously establishing the necessary infrastructure for quantum resistance. Beyond mere implementation, this structure is specifically designed to facilitate a functional evaluation and impact analysis of hybrid post-quantum cryptography. As shown in the validation flow, the process demonstrates how both classical and hybrid signatures are linked to accounts to validate transactions before they are included in a network Block. This dual-layer approach is essential for a direct comparison between legacy and hybrid schemes, enabling precise measurement of performance and resource overheads, such as key sizes, increased signature sizes, and higher computational costs associated with PQC.

Consequently, this work leverages the proposed framework to conduct a comprehensive assessment and identify potential bottlenecks through test cases. By measuring these impacts within high-performance systems like blockchain platforms, the proposal aims to establish the feasibility of hybrid post-quantum digital signatures for secure authentication and long-term data integrity in modern distributed infrastructures.

4.1. Validation Flow

The proposed architecture introduces three new attributes to the Besu transaction. First, `dsaType` is a two-byte field that specifies the verification algorithm the node must execute. Second, `publicKey` is included for algorithms that do not support PKR. Finally, `signature` is a byte sequence that replaces the traditional ECDSA `r, s, v` fields in all transaction types supporting hybrid or post-quantum signatures. To preserve backward compatibility, the four new Flexible transaction types are registered alongside the five existing types using dedicated type byte and dedicated Recursive Length Prefix (RLP) encoders and decoders, assuming that all full node have been upgraded to accept and process these new transaction types. This approach allows new and standard transaction types to coexist on the same chain without requiring historical blocks to be rewritten. A comparison of both encoding structures is illustrated in Figure 2, which contrasts the

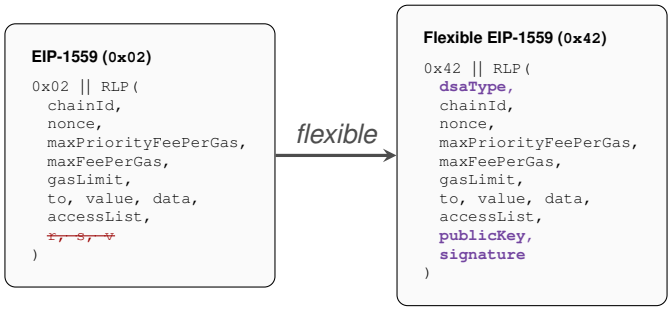


Figure 2. Side-by-side comparison of the standard (`0x02`) and Flexible (`0x42`) encodings of an EIP-1559 transaction.

standard EIP1559 transaction with its crypto-agile version, FlexibleEIP1559, that preserves all fee-market and payload fields of its standard version, with the exception of the ECDSA signature fields. EIP1559 is currently the most widely used transaction type on Ethereum, which motivated the choice of both the standard EIP1559 and its FlexibleEIP1559 as the reference transaction types for the comparative evaluation presented in Section 5.

A similar design applies to the EIP-7702 authorization object, which historically lacked a type byte. Adopting the EIP-2718 convention, authorizations now include an `AuthorizationType` byte that distinguishes a `Standard` form (`0xf8`) from a `Flexible` form (`0x01`). Mirroring the new transaction formats, this flexible variant incorporates the `dsaType`, `publicKey`, and `signature` fields. This modifications removes the ECDSA dependency from authorization objects in `FlexibleCodeDelegation`, enabling authorizations to be signed using any supported DSA. Figure 3 illustrates the six-step validation flow of the proposed hybrid-signed transaction.

1. In Step 1, the wallet uses an ECDSA-MLDSA hybrid keypair, where the secret key is defined as $\texttt{sk} = \texttt{ecdsaSk} \,\|\, \texttt{mldsaSk}$ and the public key as $\texttt{pk} = \texttt{ecdsaPk} \,\|\, \texttt{mldsaPk}$. The wallet constructs a Flexible transaction and signs it applying the Strong Nesting [Bindel et al. 2017] combiner: first, the ECDSA signature σ_1 is computed over the message; next, the MLDSA signature σ_2 is generated over $\textit{message} \,\|\, \sigma_1$. This construction preserves the strong unforgeability (SUF-CMA) guarantees of MLDSA, while a simple signature concatenation

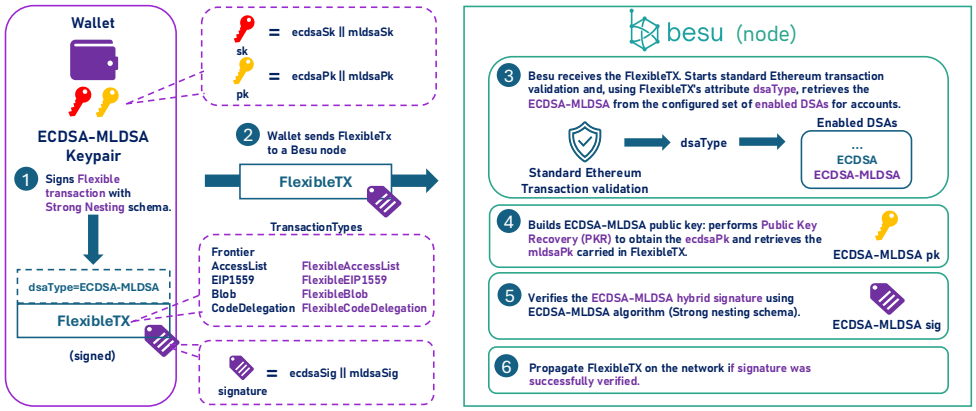


Figure 3. Validation flow of a hybrid-signed Flexible transaction.

would inherit only existential unforgeability (EUF-CMA), since ECDSA itself provides EUF-CMA security [Ghinea et al. 2023]. The final transaction signature consists of the concatenated payload $\sigma_1 \parallel \sigma_2$, with the `dsaType` field explicitly set to the ECDSA-MLDSA identifier;

2. The wallet submits the signed `FlexibleTX` to a Besu node via the standard JSON-RPC interface, concluding Step 2;
3. In Step 3, the node executes the standard Ethereum transaction validation and subsequently uses the `dsaType` field to look up the required algorithm within the configured set of enabled DSAs. This supported set is governed by a new `accounts-supported-dsas` configuration flag (e.g., `[SECP256K1, SECP256K1-MLDSA44]`), allowing network operators to control which signature schemes are accepted. This flag must be configured identically across all full node to ensure consensus integrity and synchronization errors. If the requested DSA is absent from this set, the transaction is immediately rejected;
4. In Step 4, the node reconstructs the full hybrid public key. It performs PKR to extract `ecdsaPk` from the ECDSA component of the signature and extracts `mldsaPk` directly from the `publicKey` field of the transaction. Because MLDSA does not support PKR, this explicit field transmission is essential to render the post-quantum component verifiable. Finally, these two individual components are concatenated to yield the complete hybrid public key;
5. In Step 5, the node verifies the hybrid signature using the Strong Nesting combiner, which evaluates the ECDSA and MLDSA components separately. For ECDSA, σ_1 validity is attested by the successful PKR performed in Step 4. Only σ_2 requires explicit verification, and validation succeeds if and only if σ_2 is valid against *message* $\parallel \sigma_1$ using `mldsaPk`;
6. Finally, in Step 6, upon successful verification, the node propagates the `FlexibleTX` to the rest of the network and inserts it into the mempool. Otherwise, the transaction is immediately discarded.

The address of the hybrid account is derived from the concatenated public key, establishing a cryptographic binding to both individual components. Consequently, if an adversary attempts to forge a transaction by replacing or using only one of these components, the resulting address derivation produces a distinct account address, even if the signature is successfully verified, preventing the theft or unauthorized spending of funds from the original account through the forged transaction. Thus, provided that at least one public key component is post-quantum secure, the underlying account remains quantum-safe. This approach mitigates both the account vulnerability and admin vulnerability problems [Babbush et al. 2026] by enabling users to natively migrate to accounts authenticated through a quantum-safe DSA scheme. At the same time, new contracts could start using quantum-safe administrator accounts, whereas old ones can move the administrator account to a quantum-safe address. Furthermore, to integrate a new DSA in the future, it suffices to register its parameters and the corresponding verifier implementation, subsequently activating it via the `accounts-supported-dsas` configuration flag. This design provides Besu with the necessary cryptographic agility to support a seamless post-quantum migration strategy.

5. Validation

The experiments were conducted on a Dell Latitude 5450 powered by an Intel Core Ultra 7 165U processor (Meteor Lake architecture, 12 cores, 14 threads, up to 4.9 GHz) with 32 GB of RAM, running Ubuntu Linux 24.04 LTS. The network topology is comprised of four Besu nodes orchestrated via Docker containers, using a custom image based on version 26.1.0 modified with the proposed features. These nodes operated under the QBFT consensus mechanism. To evaluate the system, both the blockchain network and the client application shared the same physical host, meaning the computational resources were distributed among the four nodes and the benchmark tool throughout all execution runs.

Cryptographic operations were implemented using the Bouncy Castle Java library (version 1.80). It is worth noting that while raw cryptographic performance benchmarks are traditionally evaluated using highly optimized C or C++ implementations to leverage low-level language efficiency, this study intentionally utilizes Java. Since Besu is natively built on the Java Virtual Machine (JVM), relying on a C-based cryptographic abstraction would mask the true systemic overhead and integration friction that occurs in actual enterprise production environments. A dedicated client application, named Flexwallet, was developed to act as both a transaction client and a benchmarking tool, implementing the hybrid ECDSA-MLDSA44 signing scheme along with the standard EIP1559 and the newly introduced FlexibleEIP1559 transaction type. The evaluation workloads consisted of 1,000 simple ether transfer transactions distributed evenly across the four nodes (250 transactions per node). Each transaction originates from a distinct sender account to ensure nonce independence, thereby eliminating transaction pool ordering as a confounding variable.

To evaluate the impact of the proposed crypto-agile architecture, the validation framework is structured into three complementary test cases, which are detailed below and assess performance from both an individual transaction perspective and a global network layer perspective: (i) authentication time; (ii) transaction throughput; and (iii) network traffic and bandwidth impact. To facilitate reproducibility and support further experimentation, all network configuration files and benchmarking scripts utilized in this evaluation are publicly available at https://doi.org/10.5281/zenodo.20347079.

5.1. Test Case #1. Authentication Time

This test case evaluates the computational latency introduced at the individual transaction level, strictly isolating the cryptographic processing time required by the standard ECDSA scheme against the proposed hybrid ECDSA-MLDSA44 signature architecture. To capture an exhaustive performance profile, authentication latency was instrumented and recorded at two distinct phases of the transaction lifecycle within the client node:

- Pool Entry Validation: Triggered immediately upon JSON-RPC submission and remote transaction propagation, where the node executes preliminary structural and cryptographic checks before admitting the transaction into the localized mempool.
- Block Validation: Executed during the block verification phase, where validator nodes must concurrently process and authenticate all compiled transactions bundled within a newly proposed block. At the current stage of implementation, signature verification is

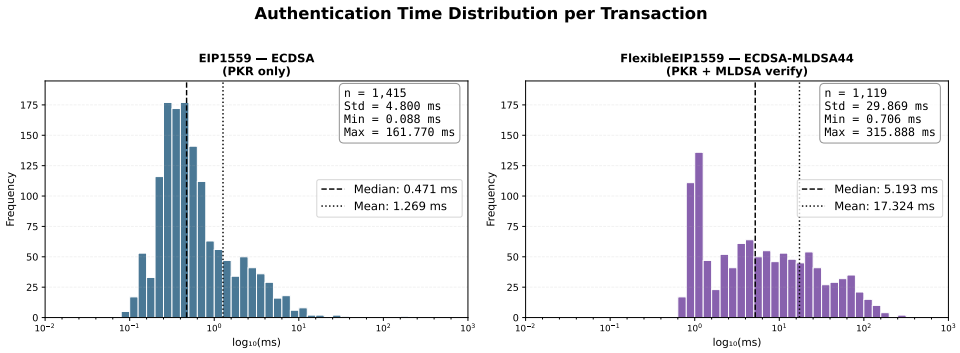


Figure 4. Authentication time distribution per transaction.

solely executed during transaction admission into the transaction pool, while, in block validation consensus phase, only PKR is performed.

For standard EIP1559 transactions, the mathematical derivation of the total authentication time corresponds to the exact average of the two PKR operations natively executed per transaction lifecycle. Conversely, for the FlexibleEIP1559 architecture, since the post-quantum MLDSA primitive lacks native support for public key recovery, the total authentication time is calculated as the cumulative sum of the classical ECDSA PKR latency and the explicit cryptographic signature verification time required by the MLDSA.

The distribution of these measurements, as illustrated in Figure 4, demonstrates a substantial performance divergence between the two paradigms. The baseline ECDSA scheme achieves a highly optimized median authentication latency of 0.471 ms. In contrast, the hybrid ECDSA-MLDSA44 scheme exhibits a median processing time of 5.193 ms, representing a computational overhead of approximately 11x. This performance delta becomes significantly amplified when analyzing the tail of the latency distribution. At the 99th percentile (P_{99}), traditional transaction authentication completes within 10.137 ms, whereas the hybrid post-quantum validation escalates to 144.764 ms translating into a 14x overhead under worst-case execution conditions.

These empirical insights indicate that while a sub-millisecond median overhead remains highly manageable for localized transaction ingestion, the pronounced spikes observed in the tail percentiles (P_{99}) pose a severe threat to system scalability. When aggregated across large and concurrent transaction propagation scenarios, this worst-case cryptographic latency compounds quadratically, increasing the cumulative CPU overhead associated with transaction admission, mempool validation and peer-to-peer transaction propagation, potentially reducing the transaction ingestion throughput under high-load conditions.

5.2. Test Case #2. Transaction Throughput

This test case examines the macro-level systemic impact of post-quantum hybrid signatures on the overall processing capacity of the blockchain network, measured in finalized transactions per second (TPS) under the QBFT consensus layer. While individual transaction ingestion at the mempool entry point remained performant for both configurations, the results reveal a severe performance degradation at the global network layer when migrating to PQC primitives.

As illustrated in Figure 5, the baseline EIP1559 configuration leveraging ECDSA sustained a network throughput of 153.1 tx/s, successfully finalizing all 1,000 transactions

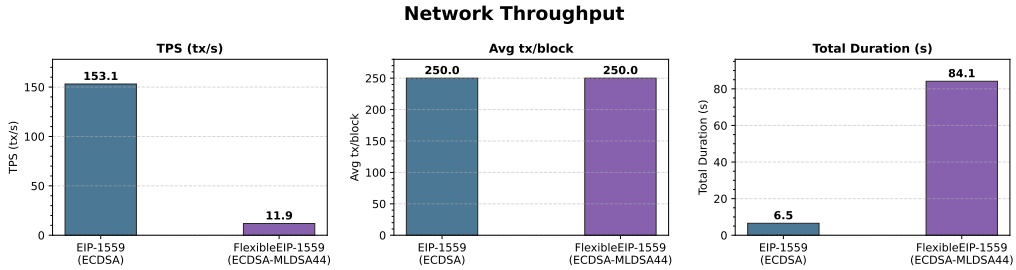


Figure 5. Comparison between ECDSA and ECDSA-MLDSA44

and achieving full block settlement in approximately 6.5 seconds. Conversely, the implementation of the FlexibleEIP1559 architecture utilizing the hybrid ECDSA-MLDSA44 scheme dropped to a global throughput of 11.9 tx/s, requiring approximately 84.1 seconds to process the identical workload. This stark contrast equates to a, approximately, 13x reduction in overall system throughput and a 13x increase in total execution time, highlighting a severe performance penalties.

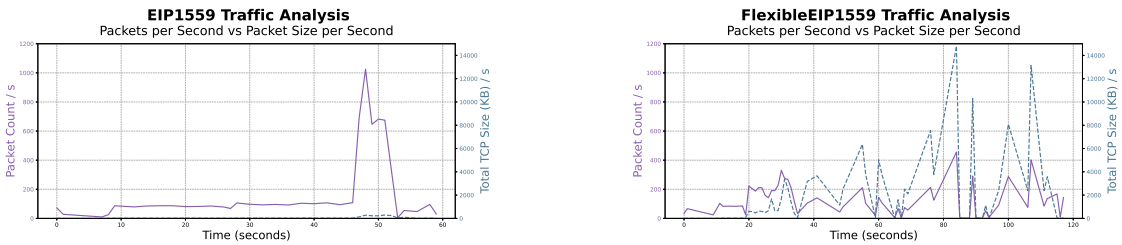
One plausible explanation for this performance degradation, considering only the experimental results, arises from two factors: the significantly larger transaction and block payloads produced by the hybrid ECDSA-MLDSA44 scheme, which forces every validator node to receive, deserialize and route oversized artifacts through rounds of the QBFT consensus protocol; and the pool entry validation stage, which imposes cryptographic verification work on every incoming transaction before mempool admission.

With a configured `blockperiodseconds` of 2 and a `requesttimeoutseconds` of 10, the maximum block gas limit that the network could structurally sustain without triggering severe round changes or consensus timeouts was 7,014,000 gas[3]. This capacity bound restricts the block size to a maximum of approximately 334 hybrid-signed transactions per block. These findings clearly demonstrate that the architectural bottleneck for post-quantum distributed ledger migration does not reside in the transaction pool mechanics, but is heavily concentrated in the cumulative computational cost of validating larger, nested cryptographic payloads at the consensus layer.

5.3. Test Case #3. Network Traffic and Bandwidth Impact

This test case evaluates the impact of larger hybrid cryptographic payloads on the network layer, specifically analyzing data propagation and bandwidth consumption during transaction broadcasting. Figure 6 presents a comparative side-by-side network traffic analysis between standard EIP1559 transactions and the proposed FlexibleEIP1559 transactions using the ECDSA-MLDSA44 hybrid scheme.

[3]Gas measures the computational effort required to execute operations on the Ethereum network.



(a) EIP1559. **(b) FlexibleEIP1559.**

Figure 6. Network Traffic Analysis.

As observed in Figure 6a, legacy EIP1559 transactions maintain a highly lightweight and stable traffic profile, due to their compact ECDSA signature format (r, s, v fields). Conversely, Figure 6b reveals a sharp, significant surge in network bandwidth usage when deploying FlexibleEIP1559 transactions. This dramatic increase in per-second data throughput is a direct consequence of the structural overhead required by post-quantum primitives. Furthermore, the traffic spikes correspond to the moments when nodes propagate newly created blocks to the peers through the network.

Specifically, while a classical transaction carries a small signature footprint, the flexible variant must transmit the explicit dsaType (2 bytes), a concatenated signature sequence ($65 + 2420$ bytes), and the uncompressed public key (1312 bytes) required to overcome the lack of PKR in MLDSA. This accumulation of metadata noticeably inflates the size of each propagated transaction payload. Consequently, although the network layer can sustain transaction routing, this substantial bandwidth expansion stresses node communication channels, compounding the delays observed during block propagation and finalization within the QBFT consensus layer.

6. Conclusion

This paper introduced a crypto-agile, hybrid post-quantum transaction architecture natively integrated into the Besu Ethereum client. By designing flexible transaction types and dedicated RLP handlers, the framework successfully accommodates hybrid ECDSA-MLDSA44 primitives while structurally mitigating the absence of native public key recovery. The empirical validation demonstrated that while localized operations within the transaction pool remain highly performant, the accumulation of larger cryptographic payloads and nested signature verifications imposes a critical computational burden on the network, explicitly delaying block finalization under the QBFT consensus mechanism and degrading global throughput. Consequently, these findings suggest that the foundational architectural bottleneck for post-quantum distributed ledgers is concentrated within the consensus layer rather than individual transaction routing.

To address these limitations, future work will investigate the integration of highly optimized, hardware-accelerated cryptographic libraries via FPGA or ASIC offloading within the JVM layer to decouple cryptographic processing from core consensus CPU cycles. In this alternative scenario, differing from the configuration adopted in the present study, each blockchain network node, including client application, will be deployed on a dedicated hardware. This modification directly addresses the Consensus-Layer bottleneck identified herein, which was amplified by both the blockchain network and the client application sharing the same physical host.

Additionally, research will focus on optimizing consensus engine dynamics through adaptive block periods or parallelized block-validation pipelines in Besu, alongside exploring state-compression techniques and zero-knowledge proofs (such as SNARKs/STARKs) to mitigate the massive network traffic overhead. Future work will also compute the gas consumption of Flexible transactions. Finally, the crypto-agile framework will be expanded to evaluate additional post-quantum signature algorithms, including NIST-standardized algorithms and candidates from the ongoing backup signature competition, such as Falcon, MAYO, SNova and SQISign, to provide a broader benchmark comparison regarding signature sizes and computational footprints in Ethereum-

based systems.

Acknowledgement

This work was supported by the Ministério da Ciência, Tecnologia e Inovação (MCTI) under Law No. 8,248 of October 23, 1991, through the PPI-SOFTEX program, coordinated by Softex, and published under PDI 19, DOU 01245.010909/2024-41.

References

Abulkasim, H. et al. (2021). Quantum-based privacy-preserving sealed-bid auction on the blockchain. *Optik*, 242:167039.

Babbush, R. et al. (2026). Securing elliptic curve cryptocurrencies against quantum vulnerabilities: Resource estimates and mitigations. *arXiv preprint arXiv:2603.28846*.

Bindel, N., Herath, U., McKague, M., and Stebila, D. (2017). Transitioning to a quantum-resistant public key infrastructure. Cryptology ePrint Archive, Paper 2017/460.

Buterin, V. et al. (2022). Eip-4844: Shard blob transactions. `https://eips.ethereum.org/EIPS/eip-4844`. Accessed: Apr 2026.

Chen, J. et al. (2021). On the construction of a post-quantum blockchain for smart city. *Journal of information security and applications*, 58:102780.

de Haro Moraes, D. et al. (2024). Applying post-quantum cryptography algorithms to a DLT-based CBDC infrastructure: Comparative and feasibility analysis. Cryptology ePrint Archive, Paper 2024/1206.

Dell, H. et al. (2014). Exponential time complexity of the permanent and the tutte polynomial. *ACM Transactions on Algorithms (TALG)*, 10(4):1–32.

Dhar, S. et al. (2024). Securing iot devices: A novel approach using blockchain and quantum cryptography. *Internet of things*, 25:101019.

Edwards, M. et al. (2020). A review of quantum and hybrid quantum/classical blockchain protocols: M. edwards et al. *Quantum Information Processing*, 19(6):184.

Fedorov, A. K. (2023). Deploying hybrid quantum-secured infrastructure for applications: When quantum and post-quantum can work together. *Frontiers in Quantum Science and Technology*, 2:1164428.

Fernandez-Carames, T. M. and Fraga-Lamas, P. (2020). Towards post-quantum blockchain: A review on blockchain cryptography resistant to quantum computing attacks. *IEEE access*, 8:21091–21116.

Fouque, P.-A. et al. (2017). Falcon: Fast-fourier lattice-based compact signatures over ntru. `https://falcon-sign.info/`. Accessed: May 2026.

Gao, Y.-L. et al. (2020). A novel quantum blockchain scheme base on quantum entanglement and dpos: Y.-l. gao et al. *Quantum Information Processing*, 19(12):420.

Garcia, C. R. et al. (2024). Quantum-resistant transport layer security. *Computer Communications*, 213:345–358.

Ghinea, D. et al. (2023). Hybrid post-quantum signatures in hardware security keys. In *International Conference on Applied Cryptography and Network Security*, pages 480–499. Springer.

Gupta, S. et al. (2023). End to end secure e-voting using blockchain & quantum key distribution. *Materials Today: Proceedings*, 80:3363–3370.

Imam, R. et al. (2021). Systematic and critical review of rsa based public key cryptographic schemes: Past and present status. *IEEE access*, 9:155949–155976.

Iovane, G. (2021). Murequa chain: Multiscale relativistic quantum blockchain. *IEEE Access*, 9:39827–39838.

Mosca, M. (2018). Cybersecurity in an era with quantum computers: Will we be ready? *IEEE Security & Privacy*, 16(5):38–41.

National Institute of Standards and Technology. Module-lattice-based digital signature standard. Technical report.

National Institute of Standards and Technology (NIST) (2024). Stateless hash-based digital signature standard. Federal information processing standards publication (fips 205), National Institute of Standards and Technology, Gaithersburg, MD, USA.

Nielsen, M. A. and Chuang, I. L. (2010). *Quantum computation and quantum information*. Cambridge university press.

Nilesh, K. and Panigrahi, P. K. (2022). Quantum blockchain based on dimensional lifting generalized gram-schmidt procedure. *IEEE Access*, 10:103212–103222.

Nouma, S. E. and Yavuz, A. A. (2024). Trustworthy and efficient digital twins in post-quantum era with hybrid hardware-assisted signatures. *ACM Transactions on Multimedia Computing, Communications and Applications*, 20(6):1–30.

Pedone, I. et al. (2021). Toward a complete software stack to integrate quantum key distribution in a cloud environment. *Ieee Access*, 9:115270–115291.

Radanliev, P. (2024). Artificial intelligence and quantum cryptography. *Journal of analytical science and technology*, 15(1):4.

Shor, P. W. (1999). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332.

State Information Center of China (2023). Blockchain-based service network (bsn): Architecture, key technologies and applications. `https://rprc.scau.edu.cn/_upload/article/files/72/6b/94a65d3f49219cebf309656790ff/ef38ed07-136c-49a8-bf71-9f2a46a8b125.pdf`. Accessed: May 2026.

Waterland, P. (2016). Quantum resistant ledger whitepaper. `https://github.com/theQRL/Whitepaper/blob/master/QRL_whitepaper.pdf`. Accessed: Apr 2026.

Yang, Z. et al. (2023). A survey and comparison of post-quantum and quantum blockchains. *IEEE Communications Surveys & Tutorials*, 26(2):967–1002.

Zeng, P. et al. (2024). Practical hybrid pqc-qkd protocols with enhanced security and performance. *arXiv preprint arXiv:2411.01086*.

Zohaib, M., Altuwaijri, F. S., and Hyrynsalmi, S. (2024). Integrating quantum computing and blockchain: Building the foundations of secure, efficient 6g technology. In *Proceedings of the 1st ACM international workshop on quantum software engineering: The next evolution*, pages 27–34.