A Case-Based Evaluation of Quantum-Resistant Data Indexing Techniques for Blockchain Databases

Asad Ali, Saima Munawar*, Nasir Naveed
Faculty of Computer Science and Information Technology, Virtual University of Pakistan, Lahore
***Correspondence**: saima.munawar@vu.edu.pk

Citation | Ali. A, Munawar. S, Naveed. N, "A Case-Based Evaluation of Quantum-Resistant Data Indexing Techniques for Blockchain Databases", IJIST, Vol. 8 Issue. 2 pp 576-596, April 2026
Received | March 07, 2026 **Revised |** April 08, 2026 **Accepted |** April 12, 2026 **Published |** April 16, 2026.

Blockchain architectures make extensive use of authenticated indexing structures, such as the Merkle Tree, for integrity and efficient retrieval of the data. However, the advent of quantum computing, in particular, through the use of Shor's and Grover's algorithms, poses a threat to classical cryptographic primitives (RSA, ECDSA, and SHA-256) on which these systems rely. While Post-Quantum Cryptography (PQC) is a solution, its integration introduces significant computational and storage overheads in terms of computation and storage, which remain underexplored in practical database environments. This research has a case study-based assessment of quantum-resistant indexing techniques in a simulated healthcare environment. This research compares a classical baseline using a private Go-Ethereum network against a custom Cryptographic Stress Test Engine modeling NIST-standardized PQC algorithms: Dilithium3 (Lattice-based), XMSS (Stateful Hash-based), and SPHINCS+ (Stateless Hash-based). Experiments were performed on a dataset of 10,000 Electronic Health Records (EHR) using resource-limited hardware to simulate a deployment in a developing infrastructure. Empirical findings show a crucial "storage explosion" in designs that are resistant to quantum computing. The classical ledger was 1.2 MB in size, the Dilithium3 increases storage by approximately 50× (1.2 MB to 62.8 MB) and SPHINCS+ by 270x (up to 325.9 MB). In terms of indexing latency, Dilithium3 was a nice middle ground option (13.10s), while SPHINCS+ exhibits higher computational cost (25.63s). The study concludes that stateless hash-based schemes are suitable for long-term archival security applications (e.g., degree verification) and lattice-based structures are more suitable for high-throughput systems, such as in real-time healthcare monitoring and national digital currencies.

Keywords: Blockchain-Based Databases, Cryptographic Data Structures, Data Indexing Strategies, Post-Quantum Cryptography, Quantum-Resilient Storage.

Introduction:

Blockchain technology has brought about a paradigm shift in the field of data management in the digital age. Established as one of the cornerstones of the new digital economy, blockchain is a reliable, decentralized, and permanent registry that does not have a central point of control [1]. As opposed to the classical Database Management System (DBMS), in which one administrator has the power to update or erase the records, blockchain spreads this power throughout the network of validators [2].

This decentralization ensures that the integrity of information, be it financial resources, supply chain documents, or cyber identities, is not secured through institutional trust, but algorithmic consensus and cryptographic verification [3][4]. The core value proposition of these systems is the fact that they can provide tamper-resistant, transparent, and auditable ledgers [5]. A simple analogy is a digital spreadsheet that has been copied thousands of times in a network of computers. Each time a new entry is suggested, it has to be authenticated by most of the network and then cryptographically connected to the past entries. Such an append-only structure guarantees that once information is stored, it cannot be modified without essentially shutting down the entire world's network [6]. Therefore, the financial and healthcare industries, among others, have integrated blockchain to get rid of intermediaries and guarantee the authenticity of data in the long term [7].

Blockchain databases are dependent on authenticated data structures to realize this scalability and integrity. Among them, Merkle Trees and Patricia Tries are the most noticeable, as they are used to perform a fast lookup and verify the state [8][9][10]. The given indexing structures enable the nodes to check the presence of a particular transaction in a huge set of data without the need to scan the complete ledger [9]. Classical cryptographic primitives (cryptographic hash functions (including SHA-256) to enforce data integrity, and Public Key Cryptography (including RSA and ECDSA), to create and verify digital signatures and to authorize transactions) are used to secure these indexes [8].

Nevertheless, the accelerated development of Quantum Computing poses an existential threat to these classical underpinnings [11]. As the process of quantum processing emerges, the core principles of contemporary blockchain security should be re-examined fundamentally. Specifically, Shor's algorithm poses a direct threat to currently used public-key signature algorithms Rivest-Shamir-Adleman (RSA) and Elliptic Curve Digital Signature Algorithm (ECDSA) because with a powerful enough quantum computer, it becomes possible to derive a private key from a public key and thus attackers can sign fraudulent transactions [12][13]. At the same time, the effective security of hash-based components is affected by the Grover Algorithm to the level that a quadratic speedup of a brute-force attack is achieved, which can compromise the integrity of Merkle Trees and other indexing structures [14][15].

Even though these threats do not immediately make classical designs obsolete, they require an active search for Quantum-Resilient or Post-Quantum Cryptography (PQC) designs. Guidelines are now being tried in the academic community, with some alternatives to Hash-based signatures, extended Merkle Signature Scheme (XMSS), Stateless Practical Hash-Indexed New Chain Signature (SPHINCS+), and Lattice-based designs (e.g., Dilithium) [16], [17][18]. These quantum-resistant options are, however, very expensive, contrary to classical methods, which are highly optimized to have good performance. They pose enormous overhead in storage needs, computation time, and complexity of implementation. In high-throughput applications, such as real-time medical data access, these trade-offs are a critical bottleneck in the implementation of the application.

This gap is filled in this research by evaluating these methods in a simulated Electronic Health Records (EHR) system using case-based assessment. This study uses a combination of experimental design instead of a purely theoretical study. Our classical baseline consists of a private Go-Ethereum (Geth) network and is compared with a custom Cryptographic Stress

Test Engine. This engine models the behavior of next-generation quantum-resistant algorithms- namely Dilithium3 (Lattice-based), XMSS (Stateful Hash-based), and SPHINCS+ (Stateless Hash-based) to measure their effect on database performance [19]. This paper quantitatively measures the exact cost of the security of storage explosion and indexing time by implementing classical and quantum-resistant indexing on a 10,000-patient record dataset in parallel. Finally, this study will offer a decision tree to prospective blockchain developers, giving empirical information on how to design future-proof and quantum-resistant databases of the most important sectors, such as healthcare.

Motivation:

This study is driven by the fact that blockchain-supported databases are gaining popularity in crucial industries, where integrity, audit records, and quick access are essential. Such systems are based on authenticated data structures like Merkle trees and digital signatures to make data verifiable at scale. Nevertheless, there is a change in the risk profile with the development of quantum computing. The algorithm by Shor poses a threat to the public key signatures, and the algorithm by Grover reduces the effective security of hash-based components.

The study is topical to an urgent challenge of the country, which is the rapid development of its own country-level blockchain infrastructure; the State Bank of Pakistan has declared preliminary work on the creation of a potential Central Bank Digital Currency (CBDC), also known as Digital Rupee, to modernize the national financial framework. Simultaneously, it looks at its concerns about the existence of regulated digital asset structures, such as the recent establishment of the Pakistan Crypto Council. In parallel, expressed interest in regulated digital asset frameworks, including the recent formation of the Pakistan Crypto Council. The country has entered a 'crypto mining phase. ' However, this dual expansion, an issuance of a digital currency and a mining of assets, provides a critical vulnerability. If this new financial infrastructure is based on the standard (non-quantum-resistant) cryptography, then both the Digital Rupee and the nation's mined reserves are vulnerable to future quantum attacks. Such adversaries might be able to break the underlying security keys and compromise the country's economic sovereignty before these digital frameworks mature.

Likewise, in the education industry, the Higher Education Commission (HEC) of Pakistan is moving to a blockchain-based degree verification system. This system aims to store records of students permanently in order to eliminate fraud and physical attestation. As university degrees are lifelong investments, their verification records should be safe for decades. In case a quantum computer cracks the blockchain that underpins these degrees in a decade or fifteen years, it would invalidate the authenticity of all of the authenticated degrees.

Thus, the rationale of the given study is not purely theoretical but practical and national. We are forced to find indexing designs that will strike the right balance between the present performance and the future security. This is to make sure that the developing digital resources of Pakistan, be it financial money or university degrees, are not jeopardized in the future.

Novelty of the study:

Although technically efficient, classical methods of indexing blockchains are mathematically susceptible to quantum attacks in the future. On the other hand, the Post-Quantum (PQ) indexing strategies that are proposed usually have bottlenecks in their performance, which makes them impractical in high-throughput settings [18]. No comparative and case-based research exists so far that quantifies trade-offs (storage, speed, complexity) of these antagonistic methods when a workload is held constant. This gap is resolved by this research, which exposes both the classical and PQ methods to a single healthcare dataset to ascertain their viability.

Research on post-quantum has accelerated, but there is a key gap. Few studies put classical and post-quantum approaches head-to-head inside blockchain databases with real workloads and measure how they behave. Most papers focus on the cryptographic strength, but leave the question of the database level in terms of query speed, storage footprint, and implementation effort partially answered [20][5]

This work addresses that gap in terms of a healthcare scenario to compare representative techniques. The goal is to assess not only security, but how well each balances efficiency and day-to-day practicality. The result is advice on the methods that are appropriate for systems that must continue to be trustworthy in a quantum era but still be able to work well at scale [5][20][19].

Research Questions:

RQ1: How do classical (e.g., Merkle Trees) and post-quantum (e.g., XMSS, SPHINCS+, Dilithium) indexing methods impact query response time and storage overheads compared to each other?

RQ2: How is the complexity of implementation of stateful hash-based schemes (such as XMSS) compared to stateless schemes (such as SPHINCS+) in a blockchain setting?

RQ3: Which is the best indexing method to use with a latency-sensitive healthcare blockchain based on the results of empirical performance?

Research Objectives:

The primary objective of this research is:

Objective 1: To compare the traditional and post-quantum indexing (Merkle Tree, XMSS, SPHINCS+, Dilithium) approaches in a blockchain-based system to manage healthcare records.

Objective 2: To measure and examine performance measures such as query efficiency (speed of indexing), storage overhead (ledger size), and verification latency.

Objective 3: To generalize findings into a decision table to inform the implementation of quantum-resistant designs in medical and other fields.

Objective 4: To use the valid simulation tools (Python, Geth) and standards (NIST PQC) to guarantee the validity of the assessment.

Literature Review:

Previous studies include extensions of authenticated blockchain structures, quantum analyses, and post-quantum alternatives. Comparative studies are conducted to contrast the efforts of Merkle and Patricia, Shor and Grover bearings, XMSS and SPHINCS+ signatures, which promote performance trade-offs [19].

Classical Indexing Techniques:

Merkle Tree:

The Merkle Tree is a cryptographic data structure that is extensively used in blockchain networks to ensure the integrity of data and efficient data verification in blockchain networks. It organizes data in the form of a binary tree, where data leaves each maintain the cryptographic hash of an individual transaction, and the internal nodes maintain the hash of their child nodes, which are joined together. This hierarchical construction makes it possible to represent the integrity of an entire data set with a single root hash (Merkle root).

One of the main benefits of Merkle Trees is the possibility to prove that a transaction was included using a Merkle proof, which only consists of a logarithmic number of hashes instead of the entire data set. This property allows lightweight clients present in the blockchain systems to verify transactions efficiently without keeping a complete record of the transactions, which makes the use of Merkle Trees a good fit for decentralized and resource-constrained environments.

In classical environments, Merkle Trees have cryptographic hash functions like SHA-2 or SHA- 3 to provide collision resistance and preimage resistance. However, in the realm of

quantum computing, Grover's algorithm brings a quadratic advantage to the brute force search, which is equivalent to taking away the margin of security of hash functions. While this does not destroy hash-based constructions, it reduces the actual strength of such constructions and requires increasing the hash size or choosing better parameters to ensure long-term security guarantees.

As a result, though Merkle Trees are structurally sound, the foundation of these trees is built on the usage of classical hashing algorithms, making these Trees an important subject of post-quantum security evaluations, especially in the context of Blockchain systems in which authenticated data indexing and long-term data integrity are important functions.

Patricia Merkle Trie:

This structure has the properties of both prefix trees and Merkle trees, and is generally called a [7]. It was first designed to optimize storage by compressing the paths in a trie, which is why it is especially efficient when working with sparse and unevenly distributed datasets. Instead of explicitly storing all the intermediate nodes, only the meaningful branches are retained, which reduces the redundancy without sacrificing the verifiability.

In the case of blockchain systems, the Patricia Merkle Trie is particularly suitable to be used for managing dynamic key-value pairs, such as the state of accounts, and to store smart contracts [21]. Each time the state is updated, a new root hash is created, although compact cryptographic proofs can still be created to prove the existence or absence of a particular key. This makes it possible to validate state transitions very efficiently on nodes without even having to store the entire dataset [21].

Ethereum-based systems usually use the SHA-3 (Keccak) hash function in the trie structure [22]. Compared to SHA-2, SHA-3 has better resistance to some cryptanalytic attacks and was designed with a different internal construction to make it more robust [5]. Nevertheless, like other hash-based authenticated data structures, the Patricia Merkle Trie is not totally safe from future quantum attacks. Grover's algorithm can theoretically weaken the effective security of hash functions, although the effect is not as serious as it is for public-key cryptography and can be partially overcome by adjusting the parameters [19].

Adelson-Velsky and Landis (AVL) Merkle Tree:

Improvement over the standard Merkle tree is achieved by the AVL Merkle Tree by making use of a self-balancing AVL structure. It ensures constant performance and stability in processing (mostly in range-based queries). Balancing is required to maintain the logarithmic height of the tree as well as optimum search times. However, just like other traditional methods, this method is based on classical cryptographic hash and is vulnerable to quantum computing attacks.

In order to build secure indexing schemes, post-quantum cryptographic systems are researched in this direction. Among them, hash-based and lattice-based methods are receiving intensive study because of their theoretical strength and practical applicability in the blockchain system [23][24].

Post-Quantum Indexing Techniques:

The following approaches are used with indexing and verifying layers in blockchain databases. XMSS and SPHINCS+ are not indices yet are digital signature schemes, and provide the kind of authenticity guarantees that are provided with Merkle-style indices in practice [25][26]. Lattice-based cryptography and hybrid designs push that toolbox a bit towards long-term, quantum resilient deployments [27].

Extended Merkle Signature Scheme (XMSS):

The Extended Merkle Signature Scheme (XMSS) is a stateful hash-based digital signature scheme that aims at offering security against quantum enemies in the long term. Hash-based signature schemes are only based on the security properties of cryptographic hash

functions, which makes them one of the most conservative solutions in post-quantum cryptography [28].

Conceptually, XMSS has a large number of one-time or a few-time signature keys that have a Merkle tree structure, where each leaf node is a one-time signature key pair, and the hash at the root of the tree is a public verification anchor. When a message is signed, a single unused leaf is used, and a path for the authentication is given, which enables the verifiers to recreate the root. This structure also allows for efficient verification while ensuring high security guarantees, as the integrity of the signatures relies on the collision resistance of underlying hash functions [29].

The main trade-off of XMSS is that it is stateful. Signers are responsible for keeping and synchronizing state information to keep track of which one-time keys have already been used because any re-use of a leaf key compromises security. This requirement brings complexity of operation in distributed or multi-validator environments like blockchains, where the chore of coordinating state across the nodes is non-trivial and may cause storage and management overhead.

As a result, XMSS is less appropriate for highly dynamic blockchain environments where signer state management is challenging. Its conservative security assumptions, along with the feasibility of enabling Merkle-tree-based indexing structures, make XMSS a practical pre-quantum equivalent for applications that are more interested in integrity and long-term trust in them, in case transaction throughput is more important than safety in the face of quantum resources.

SPHINCS+:

SPHINCS+ removes the need to keep track of the cryptographic state, while keeping the conservative security guarantees of the hash-based cryptography. Hash-based signature schemes rely on the presumed preimage and collision resistance of cryptographic hash functions, which are known, well-studied, and widely trusted primitives in the field of modern cryptography. Architecturally, SPHINCS+ is arranged in a massively large number of one-time signature keys as a multilayered hypertree data structure. At the lowest layer, the Forest of Random Subsets (FORS) scheme is used for signing message digests, while Winternitz One-Time Signatures Plus (WOTS+) is used to authenticate intermediate nodes [30] These components are recursively linked by Merkle trees, which allows for the construction of signatures without reusing one-time keys.

The stateless design makes deployment much easier in distributed and decentralized environments like blockchain, where multiple validators can sign transactions independently, and synchronization of cryptography state is impractical. By eliminating state management, SPHINCS+ prevents the type of security failure due to accidental key reuse that has been known to occur in stateful hash-based schemes like XMSS.

The tradeoff of this approach is the production of larger signature sizes and higher computational overhead during signature and verification. These costs can be a limitation in throughput on nodes that are resource-constrained and latency-sensitive applications [30]. Nevertheless, SPHINCS+ is a feasible post-quantum secure alternative that can be applied in situations where robustness, simplicity of deployment, and longevity of security are prioritized over performance. Modern research also covers hybrid approaches that include both classical and post-quantum elements to reduce performance overheads and keep them resistant to future quantum attacks [31].

Lattice-Based Structures:

Lattice schemes are hard to break due to hard problems like Learning with Errors (LWE) and Short Integer Solutions (SIS). A lattice (in cryptography terms) is a discrete regularly-spaced set of points in a multi-dimensional space, which implies the points are created through integer linear combinations of a small set of basis vectors. Although it is easy

to find a lattice, finding a short or closest vector of a high-dimensional lattice is believed to be infeasible even for quantum computers [32].

The foundations of the lattice-based cryptography were presented by Ajtai, who showed that the solution of certain average-case lattice problems can be reduced to worst-case lattice problems and gives a strong security guarantee. Later constructions like the Learning with Errors (LWE) built on this concept by introducing controlled noise to linear equations so that, computationally, it is hard to recover the secret values even if some partial information is leaked [33]. These problems are the mathematical foundation of the state-of-the-art lattice-based signature and encryption schemes, such as CRYSTALS-Dilithium and Kyber, which have been standardized by NIST as post-quantum security.

In the practical systems, lattice schemes have advantages in constructing flexible building blocks for both digital signatures and encryption, and thus can be integrated with blockchain networking, authenticated indexing, and secure storage mechanisms. However, the lattice-based designs tend to require heavier arithmetic operations and larger key sizes than the classical public key systems, affecting the amount of bandwidth and storage needed. Despite these difficulties, continuous optimization has made a big difference and improved their performance profile so that now lattice-based cryptography is a serious contender for the long-term security of blockchain databases [33].

Hybrid Dag + Lattice:

Emerging proposals combine Directed Acyclic Graph topologies with lattice-based signatures or encryptions. DAGs make parallel confirmation of transactions possible and provide a high throughput, and lattices provide quantum resistance. The combination is more aimed at providing scalability and durability, but is still experimental and often requires complex architectures and a larger number of computational resources than conventional chains.

Comparison of Indexing Techniques:

The comparison of indexing techniques is presented in Table 1, given below:

Table 1. Comparison of classical and post-quantum indexing techniques

Technique	Quantum Secure	Query Efficiency	Storage Overhead	Suitability	Related Studies
Merkle Tree	No	Fast	Moderate	Simple verification systems	[25]
Patricia Merkle Trie	No	Fast	Moderate	Key-value pair retrieval	[19]
AVL-Merkle Tree	No	Fast	High	Range-based queries	[26]
XMSS	Yes	Moderate	High	Secure logs, one-time auth	[27]
SPHINCS+	Yes	Low	Very High	Stateless signature use cases	[5][28]
Lattice-Based Trees	Yes	Fast	Moderate	Sensitive applications	[5]
Hybrid DAG + Lattice	Yes	Moderate	High	Scalable blockchain platforms	[29]

To prepare for the quantum risks, blockchain databases require more than traditional indexing. Classical structures are always attractive for their simplicity and speed, but their assurances in the long run are not guaranteed when quantum attacks are possible. Post-quantum designs such as hash tree signatures and lattice-based designs help to harden the security, yet often add to the usage of storage and computation time.

A balanced comparison, therefore, requires a similar setting in which different approaches are subjected to the same data, queries, and equipment. Through the use of real-world workloads and measurement of typical measures for scalability and performance, it is

possible for such a setting to determine the approaches that are viable in the short term and those that can be better suited to future use.

Material and Methods:

Research Design Overview:

In an effort to determine the effects of Post-Quantum Cryptography (PQC) on blockchain systems, this study used a Hybrid Experimental Approach. Complex cryptographic algorithms require theoretical mathematical models to help forecast the actual system behavior of the complex system in the real world. Consequently, the given research required a twofold approach involving a functional blockchain setup and a high-fidelity simulation engine. Figure 1 presents a visual overview of the research design and evaluation steps.

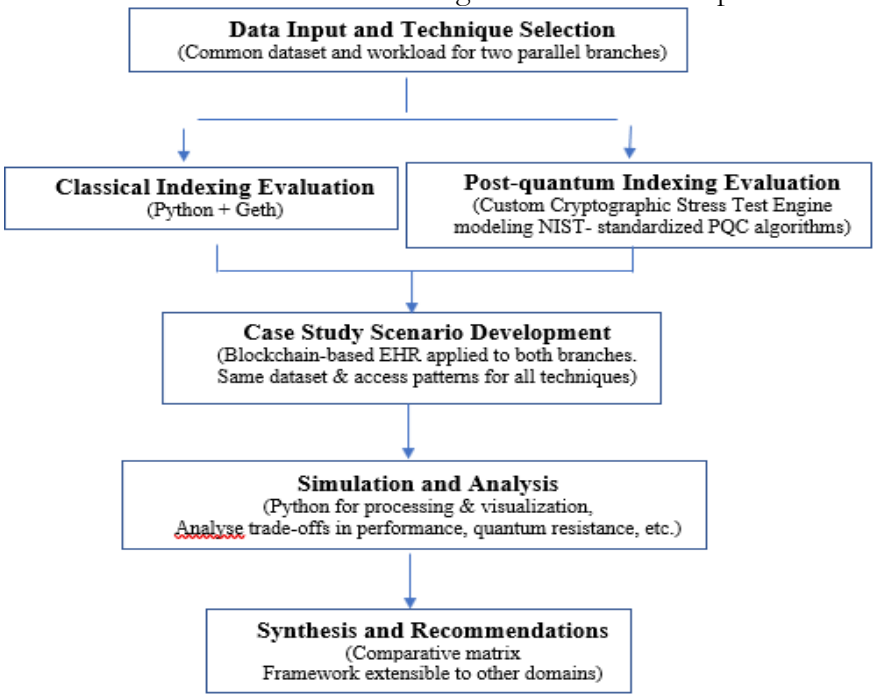

Figure 1. Methodological framework for case-based evaluation of quantum-resistant data indexing techniques for blockchain databases

The experiment design was a two-parallel design that operated on the same dataset:

The Classical Branch (Baseline):

The Go-Ethereum (Geth) was used to create a private Ethereum network. This confirmed the performance benchmark of existing industry-standard blockchain systems with classical Merkle Tree structures.

The Post-Quantum Branch (Simulation):

As a consequence of the fact that there are currently no fully developed and production-ready quantum blockchain platforms, a bespoke Cryptographic Stress Test Engine was created. Such an engine modeled the computational and storage cost of quantum-safe algorithms (NIST-standardized) (Dilithium and SPHINCS+) as though they were a part of a live blockchain index.

Case Study: Healthcare Data Management:

The healthcare sector was chosen as the main case study when conducting this evaluation. Medical records are a particularly special case of data security: they need to be confidential and authentic over decades (need to be quantum resistant), and need to be accessible instantly to clinicians (need to be query efficient). This is in line with the essential need for on-chain logs that are sensitive to privacy and auditable in the present-day healthcare infrastructure.

Dataset Generation:

A synthetic dataset made of 10,000 Electronic Health Records (EHR) was created with the Python Faker library to represent a medium-sized hospital archive. They were designed to resemble real patient files of the world, and each record was provided with:

Unique Patient ID: An indexing primary key.

Timestamp: Indicating the time when the patient was visited.

Diagnosis Codes: Reflecting confidential medical information.

Cryptographic Signature: This is a digital signature that is used to confirm the authenticity of the entry.

Experimental Setup and Equipment:

One of the goals of this paper was to show that quantum-resistant blockchains could be implemented on regular, commercially off-the-shelf hardware and not necessarily on high-performance computing clusters (HPC).

Hardware Specifications:

All experiments were performed on a mid-range workstation with the following specifications:

Processor: Intel Core i5 (Standard Workstation CPU)

Memory (RAM): 8 GB DDR3

Storage: Hybrid Configuration (256 GB SSD Primary + 500 GB HDD Secondary)

Rationale behind Resource Limitations: The choice of 8GB of DDR3 RAM was not random. It was meant to be used in environments with limited resources that are typical of health care systems, like an older hospital terminal or an IoT edge device. If the suggested Post-Quantum algorithms are effective on this hardware, then they will also be viable on the enterprise-grade servers.

Classical Configuration (Control Group):

In the case of the baseline performance metrics, the study used Go-Ethereum (Geth v1.16.7). The network was set up as a private Proof-of-Work (PoW) network. In order to control the variable of indexing speed, the mining difficulty was reduced to the lowest value, giving the transaction throughput more priority than the mining depth. This setup gave a true-to-life simulation of present-day Ethereum enterprise environments.

Data Collection Protocol:

A Python automation program written in custom Python and made with the Web3.py library was used to send data to the Geth node and measure throughput with a very high level of accuracy. The protocol used to measure was as follows:

Capture the UNIX timestamp (TStart) as soon as the first transaction submission has been made.

Send 10,000 transactions to the transaction pool of the Geth node.

Monitor block height until all transactions are mined and verified.

Record the timestamp (TEnd) upon confirmation of the 10,000th transaction.

Calculate Transactions Per Second (TPS) using the formula:

$$TPS = (\text{Total Transactions}) / ((T\_End - T\_Start))$$

Visual documentation of the experimental setup, including dataset generation and the initialization of the private Go-Ethereum node, is provided in Appendix A.

Post-Quantum Simulation Details:

In the case of the experimental group, a Python-level Cryptographic Stress Test Engine had been created. This engine has built Merkle Trees over three different security profiles with the most recent standards of the NIST Post-Quantum Cryptography.

To emulate the computational overhead of post-quantum cryptographic operations, a CPU_Burn function was implemented within the simulation environment. This function artificially consumes processor cycles to approximate the time complexity of signature generation and verification for each PQC algorithm. The intensity of CPU_Burn was

calibrated based on the relative computational cost reported for Dilithium3, XMSS, and SPHINCS+. Indexing latency was simulated by measuring the total execution time required to process each transaction, including cryptographic signing, verification, and indexing operations. The latency values were averaged over multiple runs to ensure consistency and reduce measurement variance.

The storage overhead of each PQC algorithm was estimated according to its standardized signature and key size. As an example, Dilithium3 generates signatures of around 2.7 KB, XMSS signatures lie between 2-4 KB depending on settings, and SPHINCS+ signatures can be greater than 8 KB. The simulation added these sizes to compute the total storage spent on blockchain transactions, including index structures and metadata. A comparative study of the storage requirements of classical and post-quantum indexing methods was conducted by summing these values across the dataset.

Lattice-Based (DILITHIUM3):

Chosen as the "Moderated Challenger." Lattice schemes have been observed to have a good balance between speed and security and are thus applicable to trusted data management infrastructures. Even though they are efficient, they have larger key sizes compared to classical methods.

Stateful Hash-Based (XMSS):

Chosen as the Conservative Alternative. XMSS can be applied to resource-constrained systems such as smart grids. It is based on standard hashing, but must have complex state management to support the use of one-time signatures.

Stateless Hash-Based (SPHINCS+):

Selected as the "Heavy Lifter." Being a stateless scheme, it is easier to implement but produces much larger signatures. This algorithm was added in order to investigate the maximum storage overhead.

In order to achieve reproducibility, the essence of the stress test engine is outlined in detail in Algorithm provided below. This algorithm explains the way in which the simulation attempts to compute the patient records, uses the particular cryptographic overhead established by the NIST security profile, and builds the resulting Merkle Tree.

```
Initialize List Leaves = []
FOR each Record R in D:
H = SHA-256(R)
# Simulation of PQC Overhead
Signature = GenerateRandomBytes(P.SignatureSize)
CPU_Burn(P.VerificationLatency)
Node = CreateNode(Hash=H, Sig=Signature)
Leaves.append(Node)
END FOR
WHILE length(Leaves) > 1:
NextLayer = []
FOR i = 0 to length(Leaves) step 2:
Left = Leaves[i]
Right = Leaves[i+1]
ParentHash = SHA-256(Left.Hash + Right.Hash)
NextLayer.append(ParentHash)
END FOR
Leaves = NextLayer
END WHILE
Return Leaves[0] (Root)
```

Algorithm: Cryptographic Stress Test Logic

Input: Dataset D (10,000 Records), Algorithm Profile P
Output: Merkle Tree Root Hash, Indexing Time T, Storage Size S
Results:

This section presents the experimental results in relation to the research objectives defined in Section 1.2. Specifically, the comparison of indexing techniques (Objective 1) and evaluation of performance metrics such as latency and storage overhead (Objective 2) are analyzed, while the findings contribute to the decision framework outlined in Objective 3.

Classical Baseline Performance:

The classical performance on the baseline is presented below:

To have a reliable performance baseline, 10,000 patient records were ingested in repeated experimental trials. The system showed an average throughput of 30.95 Transactions Per Second (TPS), and the observed system performance fluctuated between 26.82 TPS and 35.08 TPS depending on the background CPU load of the host machine. The cumulative confirmation rate of transactions is shown in Figure 2. The shaded area shows the variance of performance, which is common to Proof-of-Work (PoW) mining using non-dedicated hardware.

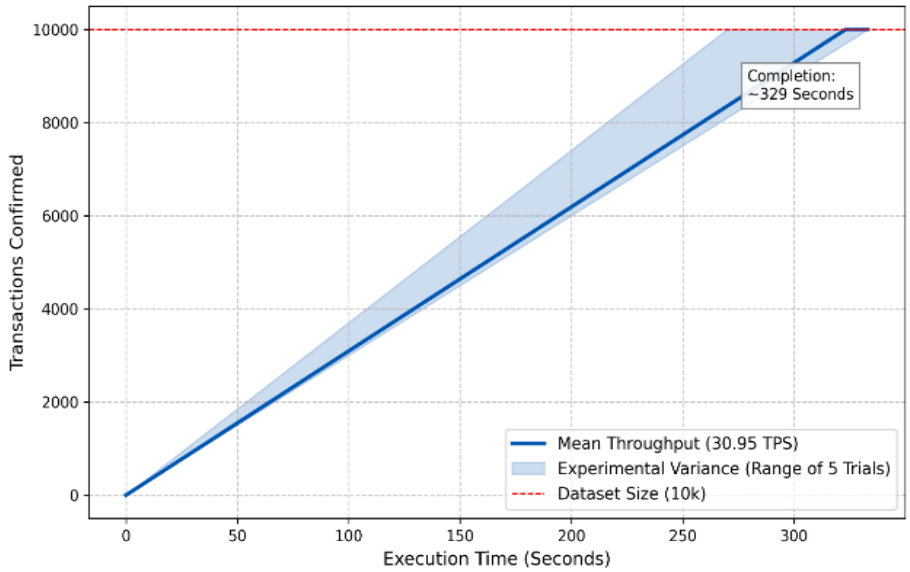

Figure 2. Classical Blockchain Indexing Performance

Fastest run completed in 285 seconds (~35 TPS) and slowest run completed in 373 seconds (approx. 27 TPS). The mean time to verify the complete 10000 record data was 329 seconds (approx. 5.5 minutes).

This variance is a confirmation that even if Geth has robust properties, its throughput performance on one node is highly dependent on stochastic mining intervals. The baseline of ~31 TPS is the control measure against which the Post-Quantum algorithms will be assessed compared to it. Screenshots of the terminal execution logs, demonstrating the live transaction processing and the final throughput metrics obtained during the benchmarking phase, are documented in Appendix A.

Storage Overhead Analysis:

The most notable observation of this work is the drastic change in storage demands in the case of switching to quantum-resistant security.

Classical Index (ECDSA):

The ledger size was minimal as it occupied 1.2 MB.

Figure 3 illustrates the comparative storage footprint for the 10,000-record index:

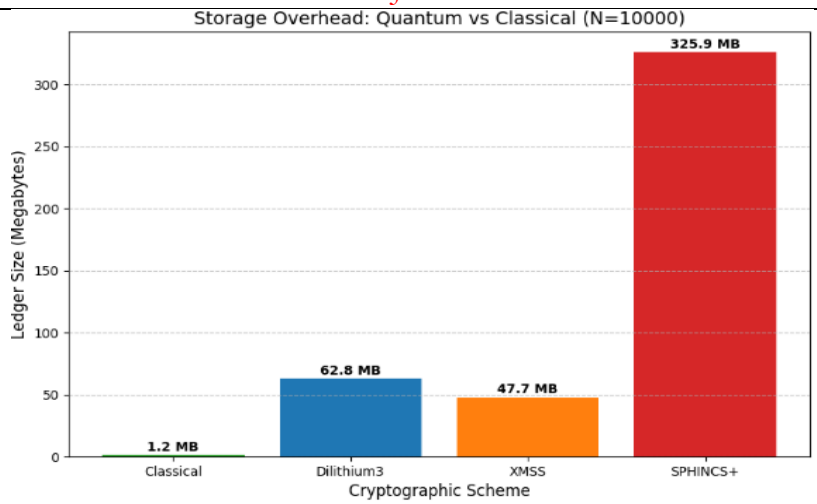

Figure 3. Comparative storage footprint for the 10,000-record index

This small size enables the entire index to be installed in the RAM of low-power devices.

Dilithium3 (Lattice):

The ledger size had gone up to 62.8 MB. This is a 50-fold increment in storage overhead. Although this is quite a significant leap, it is still within the manageable scope of the current SSD-equipped servers.

SPHINCS+ (Stateless Hash):

Its ledger size had gone to 325.9 MB. This is a 270 times increase compared to the classical level.

These findings suggest that Lattice-based approaches have a controllable cost, and stateless hash-based approaches result in a storage overhead that can be extremely costly to deploy on limited hardware.

Indexing Efficiency (Query Speed):

In addition to storage, the time it took to cryptographically verify and construct the entire Merkle Tree was also measured in the study as shown in Figure 4. This measure is a direct proxy of Query Efficiency because verifying a record takes the same mathematical operations as indexing the record.

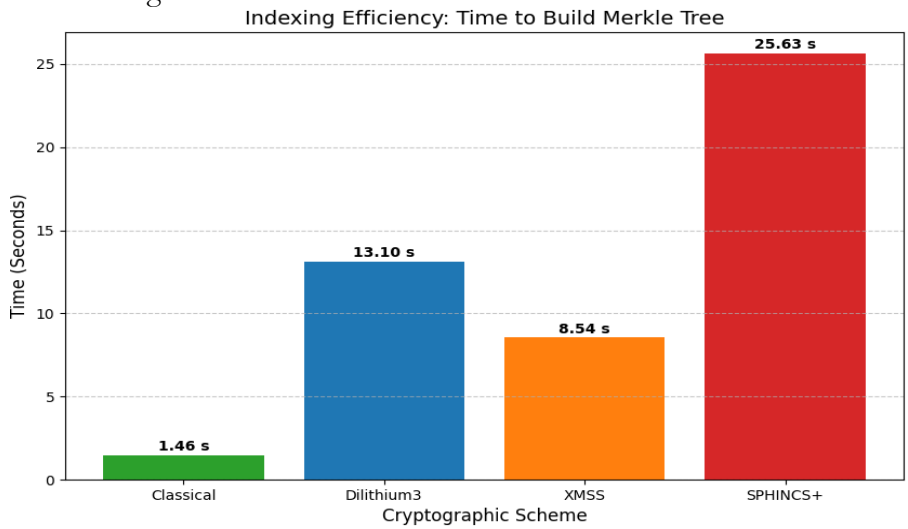

Figure 4. Presents the time-to-index results:

Classical Indexing:

Took 1.46 seconds to complete. This performance accounts for the common use of Merkle Trees in modern systems.

Dilithium3:

Time taken = 13.10 seconds. This is close to 9 times slower than the baseline. Nevertheless, a latency of 13 seconds when updating a large batch is regarded as acceptable in the asynchronous medical setting.

SPHINCS+:

Time taken to complete is 25.63 seconds. This is an almost 18 times slower rate compared to the baseline and twice as slow compared to Dilithium3. The stateless scheme imposed heavy work on hashing operations, which created an apparent processing bottleneck.

Summary of Findings (Decision Matrix):

The decision matrix with respect to the organizations that intend to upgrade their PQC can be developed based on the empirical data obtained in the stress tests. Table 2 (Decision Matrix) summarizes the comparative performance of classical and post-quantum indexing techniques based on key evaluation criteria along with their suitability for various systems. It provides a structured and practical framework for selecting appropriate algorithms depending on system requirements such as latency, storage, and security. This matrix allows system designers to choose the right indexing methods depending on application-specific needs, like real-time processing, storage limitations, or long-term integrity of data.

Table 2. Decision Matrix for Selecting PQC Indexing Techniques

Criteria	Merkle Tree (Classical)	Dilithium (Lattice-based)	XMSS (Hash-based)	SPHINCS+ (Hash-based)
Quantum Resistance	Not Resistant	Resistant	Resistant	Resistant
Storage Overhead	Low (~1.2 MB)	Moderate (~62.8 MB)	Moderate	Very High (~325.9 MB)
Indexing Latency	Very Low (~1.46 sec)	Moderate (~13.10 sec)	Moderate	High (~25.63 sec)
Query Efficiency	High	Moderate	Moderate	Low
Suitability for Real-time Systems	Excellent	Good	Limited	Poor
Suitability for Archival Systems	Not Suitable	Moderate	Good	Excellent
Implementation Complexity	Low	Moderate	Moderate	High
Recommended Use Case	Legacy/Non-secure systems	Healthcare and Fintech systems	Secure logging systems	Long-term archival systems

The results indicate that Dilithium is suitable for real-time systems, and SPHINCS+ is a better option for long-term archival applications requiring stronger security guarantees.

Discussion:

The Security-Performance Trade-Off:

The experiments reveal a basic discrepancy in the architecture of future blockchain-based systems; the need to achieve long-term quantum resistance is directly opposite to the need to achieve efficiency in the system. The findings suggest that the performance cost of perfect security (as provided by conservative hash-based schemes such as SPHINCS+) makes it impractical in high-throughput and real-time applications.

Implications of Storage Explosion:

Previous Sections show that, as theorized in earlier studies, PQC signatures have the potential to jam blockchain networks. An increase of the ledger size (SPHINCS+) 270 times would soon saturate the storage capacity of edge devices and mobile health terminals.

But the 50x change in the case of Dilithium3 presents a way out. Although 62.8 MB is much bigger than 1.2 MB, the contemporary healthcare infrastructure is based on servers with extensive storage of SSDs. Hence, the storage overhead of Lattice-based cryptography is a worthwhile expense in guaranteeing the integrity of patient data. This result confirms the

recent literature that Lattice-based systems are the most viable solution to trusted data management.

Latency and Real-Time Access:

Time is of the essence in hospitals. A physician retrieving the history of a patient cannot afford to spend a long time confirming it. The verification time of the classical system is 1.46 seconds, which is perfect, but sensitive. Another alternative that is slow but effective is dilithium3 (13.10s).

Although not immediate, it works within reasonable boundaries of batch processing. SPHINCS+ (>25s) had many troubles with the test hardware. The mathematical strength it takes to authenticate one record indicates that such a scheme would introduce intolerable delays in a busy hospital network.

Structural Considerations: Merkle Vs. Patricia Tries:

It is necessary to mention that the PQC simulations in this study were done using a Standard Binary Merkle Tree. In practice, Ethereum networks use the more complicated Patricia Merkle Trie, with extra overhead of branch nodes and path coding. Thus, the storage bloat, which is seen here (50x - 270x), can be viewed as a lower conservative estimate. An implementation of PQC into a Patricia Trie would probably also push the storage requirements even higher because the larger quantum keys would have to be stored at a variety of different tiers of the trie structure.

Conclusion On Standards Alignment:

These findings are not just theoretical simulations, but the correct previews of the future industry standards, because the NIST FIPS 204 compliant algorithms (Dilithium and SPHINCS+) are used. The conclusion is obvious: healthcare blockchains can be quantum-resistant, but the hardware infrastructure should be modified to handle the increased computational load that was observed in this study.

The experimental results also highlight the trade-offs that are practically relevant in implementing post-quantum indexing methods. The measured storage overhead from 1.2 MB in the classical implementation of Merkle Tree to 62.8 MB in Dilithium3 and 325.9 MB in SPHINCS+ points out the huge resource demands that quantum-resistant schemes will require. Similarly, the more verification latency of 1.46 seconds in the classical model, as compared to 13.10 seconds with Dilithium3 and more than 25 seconds with SPHINCS+, can be interpreted as the computational complexity of these algorithms. In spite of these overheads, the findings show that lattice-based solutions are a viable trade-off between security and performance, and hash-based solutions are more appropriate for long-term data integrity or archival purposes. The observations support the fact that cryptographic methods in blockchain systems have to be implemented contextually.

Conclusion and Future Work:

Summary of Research:

The imminent emergence of quantum computing is an existential threat to the cryptography of current blockchain systems. Although the algorithms by Shor and Grover will break the security of ECDSA and SHA-256, respectively, the post-quantum cryptography (PQC) would incur serious performance costs that have not been sufficiently quantified in real-world database settings.

This research filled the gap between theoretical cryptography and practical database management using a Hybrid Experimental Approach. This paper quantified the precise cost of security of healthcare data by benchmarking a classical Ethereum private network to a simulated Quantum-Resistant Indexing Engine. The study was able to achieve its goals of determining the trade-offs in the particular case of storage overhead, query latency, and long-term security.

Key Findings:

Three major conclusions of this study are based on the empirical results:

The "Storage Explosion" is Real

The change of the classical indexing to the stateless SPHINCS+ scheme led to an increase in the ledger size by 270x (1.2 MB vs. 325.9 MB). This proves that the ideal security is already too resource-heavy to be achieved by conventional edge devices or mobile health terminals.

Lattice-Based Cryptography is the Viable Middle Ground:

The Dilithium3 scheme provided a balanced alternative, which doubled storage (62.8 MB) and indexing time (13.1s) by 9x and 50x, respectively. Although they are heavier than the classical methods, they are within the operational capabilities of the current-day hospital servers with SSD storage.

Table 3 provides a comparison of classical and post-quantum indexing techniques in terms of performance and security. Moreover, the suitability of these techniques for various cases is also recommended.

Table 3. Performance and security of Classical and post-quantum indexing techniques, along with recommendations for use cases

Algorithm	Security	Performance Cost	Recommendation
ECDSA (Classical)	Vulnerable to Quantum	Low (1.2 MB / 1.5s)	Phase out immediately
Dilithium (Lattice)	Secure	Medium (62 MB / 13s)	Best choice for general use
SPHINCS+	Secure (Very High)	High (326 MB / 26s)	Use only for Archival Storage

Hardware Is the Bottleneck:

The tests on 8GB RAM have proven that software upgrades are not enough. PQC needs a simultaneous hardware infrastructure upgrade to sustain the real-time query rates that clinicians demand.

Implications of the Study:

These findings are discussed in the motivation, relevant to the emerging digital ecosystem. There are several implications for researchers, policy makers, and industrialists in developing a secure post-quantum blockchain system for the future.

The State Bank should consider adopting Lattice-based designs (such as Dilithium) rather than the use of the conservative hash-based schemes to make the national digital currency fast enough to facilitate retail transactions.

The blockchain of the Higher Education Commission can spare the increased latency of SPHINCS+ for use in archiving. As the process of verifying a degree is a rare occurrence (as opposed to daily payments), the greater long-term security of SPHINCS+ is worth the increased computational cost, as degrees will be resistant to tampering over decades.

For blockchain developers and system architects, the results suggest that lattice-based algorithms such as Dilithium3 provide better security and performance, which makes them suitable for real-time applications. On the other hand, hash-based schemes like SPHINCS+ are more appropriate for systems where long-term security is required over performance, such as archival storage.

Recommendations:

Table 4 gives a detailed insight into recommendations from the findings of the study.

Table 4. Recommendation as per empirical findings

Research Objective	Key Empirical Findings	Specific Actionable Recommendation
Objective 1 & 2: Compare traditional vs. post-quantum indexing and measure performance costs.	The "Storage Explosion" is real; SPHINCS+ increased ledger size by 270x (1.2 MB to 325.9 MB) while Dilithium3 increased it by 50x.	General Recommendation: Organizations must prioritize Lattice-based designs (Dilithium3) for high-throughput systems to avoid saturating storage on edge devices.
Objective 2: Examine query efficiency and verification latency.	Indexing time for SPHINCS+ (25.63s) is nearly 18x slower than the classical baseline (1.46s), creating an "intolerable delay" for real-time environments.	Infrastructure Recommendation: Upgrading to PQC requires a simultaneous hardware infrastructure upgrade (specifically SSDs and increased RAM) to sustain the query rates clinicians' demand.
Objective 3: Generalize findings into a decision table.	Different algorithms suit different sectors based on their sensitivity to latency vs. long-term security needs.	Policy Recommendation: Use SPHINCS+ for the HEC degree verification system where archival security is more critical than daily transaction speed. Dilithium3 is recommended for real-time applications.
Objective 4: Use valid simulation tools and NIST standards.	NIST FIPS 204 compliant algorithms (Dilithium and SPHINCS+) provide reliable previews of future industry standard behavior.	Technical Recommendation: Future implementations should investigate Hardware Acceleration (FPGAs) to offload heavy hashing and reduce verification delays.

Moreover, system designers are advised to be context-sensitive in determining the indexing method, with the cryptographic scheme dynamically adjusted to application requirements in terms of frequency of transactions, storage limitations, and sensitivity of security. This guarantees maximum use of post-quantum cryptographic techniques in various blockchain-based systems.

Limitations:

There were certain limitations in this study:

A Python stress-test engine was used to simulate the Post-Quantum branch as opposed to a fully re-engineered Ethereum kernel. Although the mathematical overhead is correct, network propagation delays have not been modeled.

The 8GB RAM size makes the inference to high-performance computing (HPC) clusters more difficult, but it is more realistic in the framework of the developing countries' infrastructure.

Future Work:

The subsequent studies must be aimed at optimizing the implementation of the following heavy algorithms:

Hardware Acceleration:

Investigating the use of Field-Programmable Gate Arrays (FPGAs) to offload the heavy hashing operations of SPHINCS+, potentially reducing the 25-second verification delay.

Layer-2 Off-Loading:

It may be interesting to investigate architectural designs in which the only data (the "Merkle root") is placed on the costly quantum-safe blockchain and the bulk of the signatures are stored on cheaper, off-chain decentralized storage (IPFS).

Hybrid Transition Models:

Developing protocols that allow a blockchain to accept both classical and quantum signatures simultaneously during a multi-year transition period.

Author's Contribution: All Authors contributed towards idea refinement. Author 1 contributed to the paper write-up, literature, research implementation, and reference management. Authors 2 and 3 reviewed the whole work.

Conflict of Interest: There exists no conflict of interest for publishing this manuscript in IJIST.

Appendix A:

Experimental execution and tooling:

This appendix provides visual validation of the experimental environment, data generation, and benchmarking processes applied to evaluate the indexing structures within the private blockchain network.

Figure A.1. Execution of the Python synthetic data generation script, verifying the creation of the 10,000 JSON patient records, which have been used as the standard dataset.

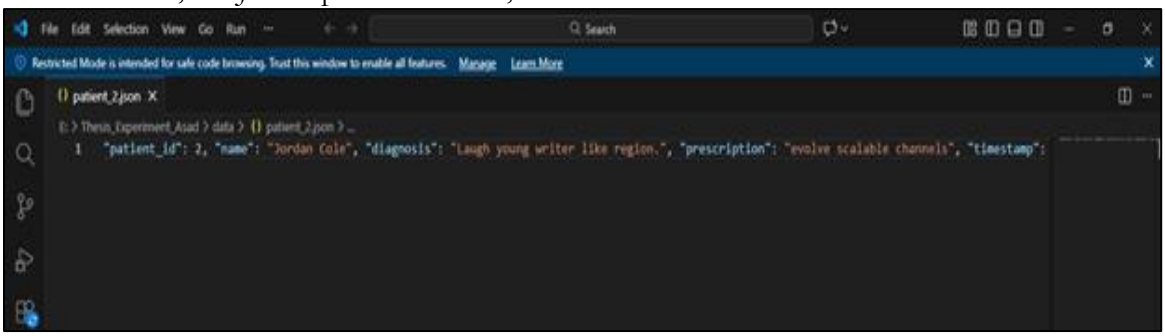

Figure A.2. Contents of each data node

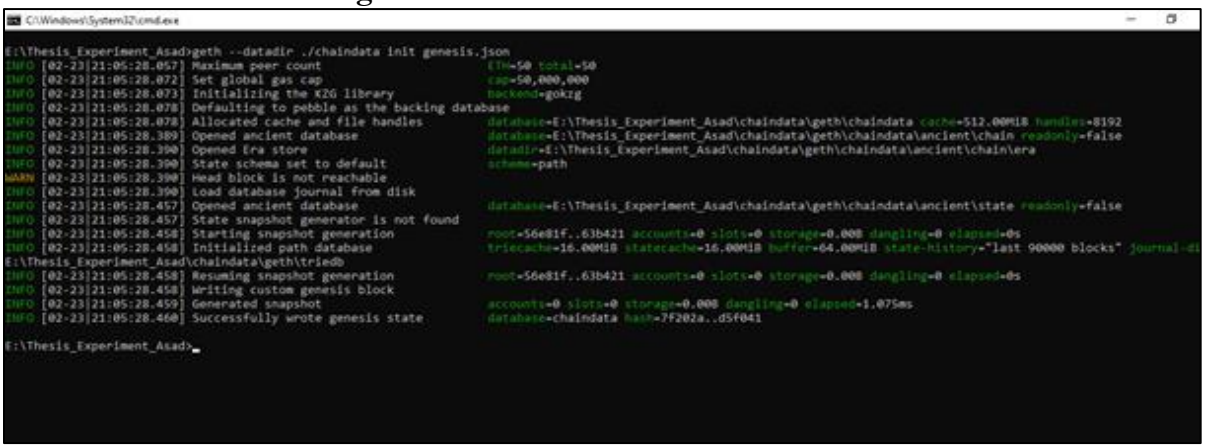

Figure A.3. Initialization of Genesis state for private Go-Ethereum (Geth) node

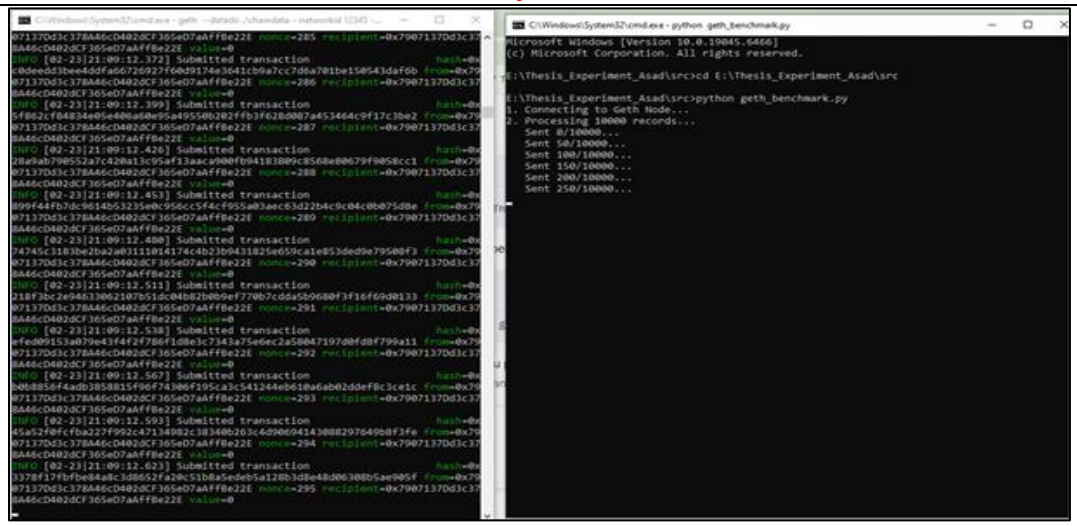

Figure A.4. Active execution of the private Go-Ethereum (Geth) node in mining mode. 10,000 records sent to a private Go-Ethereum (Geth) node by writing custom Python code utilizing Python's web3.py

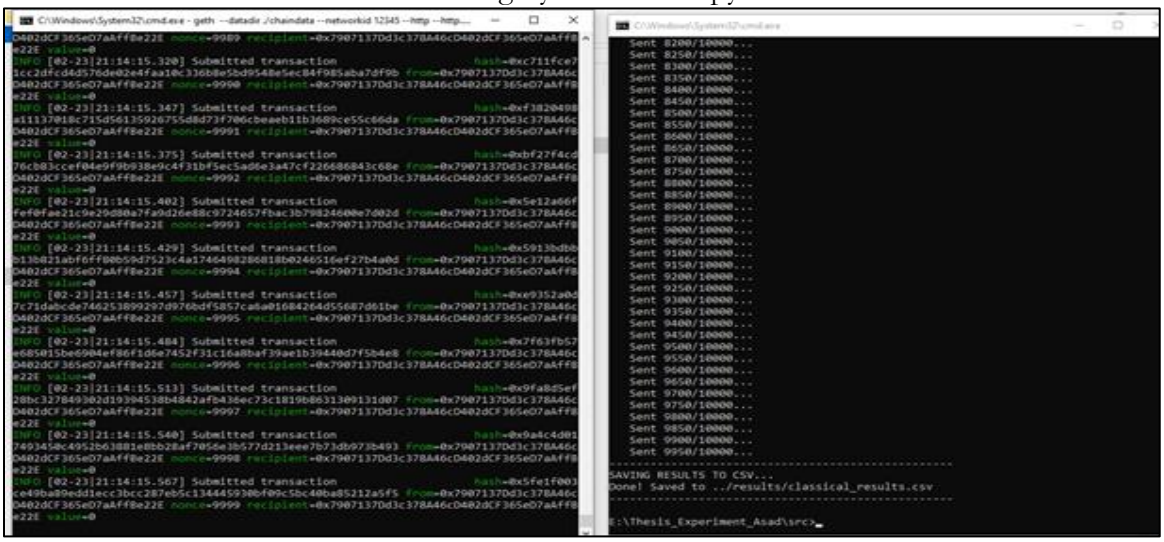

Figure A.5. Live execution logs of the benchmarking script pushing transactions to the Geth node, demonstrating the transaction flow

Metric	Value
Total Records	10000
Total Time (Seconds)	311.3565
Throughput (TPS)	32.12
Platform	Geth (Proof of Work)

Figure A.6. The final resulting throughput metrics (classical_results) of a single experiment

	Algorithm	Query Time (s)	Storage (MB)
2	Classical (SHA-256)	103.6943345	1.220703125
3	Dilithium3 (Lattice)	21.41443992	63.11416626
4	XMSS (Stateful)	14.46267867	47.68371582
5	SPHINCS+ (Stateless)	37.64654016	325.9277344
6			

Figure A.7. Running of Post-Quantum Cryptography (PQC) stress test script (pqc_stress_test.py). The terminal output shows the relative indexing time and storage (RAM) overhead of indexing 10,000 synthetic patient records using Classical (SHA-256) cryptographic profile, Dilithium3, XMSS, and SPHinCS+ cryptographic profiles.

References:

[1] M. A. Mansoor, M. Ali, A. Mateen, M. Kaleem, and S. Nazir, "Blockchain Technology for Land Registry Management in Developing Countries," *2023 2nd Int. Conf. Emerg. Trends Electr. Control. Telecommun. Eng. ETECTE 2023 - Proc.*, 2023, doi: 10.1109/ETECTE59617.2023.10396736.

[2] Sabreen Ahmadjee, Carlos Mera-Gómez, "A Study on Blockchain Architecture Design Decisions and Their Security Attacks and Threats," *ACM Trans. Softw. Eng. Methodol.*, vol. 31, no. 2, pp. 1–45, 2022, [Online]. Available: https://dl.acm.org/doi/10.1145/3502740

[3] Shi Dong, Khushnood Abbas, "Blockchain technology and application: an overview," *PeerJ Comput. Sci.*, vol. 9, 2023, [Online]. Available: https://www.researchgate.net/publication/376051976_Blockchain_technology_and_application_an_overview

[4] G. M.Faruk Ahmed, Imran Hasan, "Enhancing e-KYC Security and Privacy: Harnessing Quantum Computing and Blockchain in Web 3.0," *Distrib. Ledger Technol.*, vol. 3, no. 4, pp. 1–23, 2024, [Online]. Available: https://dl.acm.org/doi/full/10.1145/3686166

[5] Tiago M. Fernandez-Carames, Paula Fraga-Lamas, "Towards Post-Quantum Blockchain: A Review on Blockchain Cryptography Resistant to Quantum Computing Attacks," *IEEE Access*, vol. 8, 2020, [Online]. Available: https://ieeexplore.ieee.org/document/8967098

[6] Shuyun Shi, Debiao He, "Applications of blockchain in ensuring the security and privacy of electronic health record systems: A survey," *Comput. Secur.*, 2020, [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/32834254/

[7] D. Commey, S. G. Hounsinou, and G. V. Crosby, "Post-Quantum Secure Blockchain-Based Federated Learning Framework for Healthcare Analytics," *IEEE Netw. Lett.*, vol. 7, no. 2, pp. 126–129, 2025, doi: 10.1109/LNET.2025.3563434.

[8] Pooja Dhiman, Santosh Kumar Henge, "Blockchain Merkle-Tree Ethereum Approach in Enterprise Multitenant Cloud Environment," *Comput. Mater. Contin.*, vol. 74, no. 2, pp. 3297–3313, 2022, doi: https://doi.org/10.32604/cmc.2023.030558.

[9] H. Liu, X. Luo, H. Liu, and X. Xia, "Merkle Tree: A Fundamental Component of Blockchains," *2021 Int. Conf. Electron. Inf. Eng. Comput. Sci. EIECS 2021*, pp. 556–561, Sep. 2021, doi: 10.1109/EIECS53707.2021.9588047.

[10] Ralph C. Merkle, "A Certified Digital Signature," *Adv. Cryptol. — CRYPTO' 89 Proc.*, 2001, [Online]. Available: https://link.springer.com/chapter/10.1007/0-387-34805-0_21

[11] Rongjie Zhou, Huaqun Guo, "Investigating Post-Quantum Cryptography to Secure Transmitted Data via Mobile Communication," *Electronics*, vol. 15, no. 6, p. 1257, 2026, doi: https://doi.org/10.3390/electronics15061275.

[12] S. Mishra, O. Agarwal, and S. K. Patel, "Quantum Decryption using Shor's Algorithm," *2024 IEEE Pune Sect. Int. Conf. PuneCon 2024*, 2024, doi: 10.1109/PuneCon63413.2024.10895777.

[13] S. K. Shandilya, C. Ganguli, A. Kumar, I. Izonin and M. Gregus, "Post-Quantum Cryptography and Nature-Inspired Cyber Defense: Strategic Readiness and Adaptive Techniques for Next-Gen Threat Response," *IEEE Access*, vol. 14, pp. 27418–27434, 2026, doi: 10.1109/ACCESS.2026.3663832.

[14] L. K. Grover, "A fast quantum mechanical algorithm for database search," *Proc. Annu. ACM Symp. Theory Comput.*, 1996, [Online]. Available: https://dl.acm.org/doi/10.1145/237814.237866

[15] Richard Preston, "Applying Grover's Algorithm to Hash Functions: A Software

Perspective," *arXiv:2202.10982*, 2022, [Online]. Available: https://arxiv.org/abs/2202.10982

[16] Daniel J. Bernstein, Daira Hopwood, Andreas Hülsing, Tanja Lange, Ruben Niederhagen, "SPHINCS: Practical Stateless Hash-Based Signatures," *Adv. Cryptol. -- EUROCRYPT 2015*, 2015, [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-662-46800-5_15

[17] J. Di, T. Xie, S. Fan, W. Jia, and S. Fu, "An Anti-Quantum Signature Scheme over Ideal Lattice in Blockchain," *Proc. - 2020 Int. Symp. Comput. Eng. Intell. Commun. ISCEIC 2020*, pp. 218–226, Aug. 2020, doi: 10.1109/ISCEIC51027.2020.00054.

[18] M. Ma, X. Ji, and J. Cai, "A Configurable XMSS Post-Quantum Hardware Implementation with SM3 and SHA-256," *2024 4th Int. Conf. Commun. Technol. Inf. Technol. ICCTIT 2024*, pp. 220–225, 2024, doi: 10.1109/ICCTIT64404.2024.10928459.

[19] "Status Report on the First Round of the Additional Digital Signature Schemes for the NIST Post-Quantum Cryptography Standardization Process | NIST." Accessed: Mar. 31, 2026. [Online]. Available: https://www.nist.gov/publications/status-report-first-round-additional-digital-signature-schemes-nist-post-quantum

[20] S. Abbas, A. Sultana, and G. Kaddoum, "Quantum-Safe Blockchain in Hyperledger Fabric," *IEEE Netw. Lett.*, vol. 7, no. 1, pp. 61–65, 2025, doi: 10.1109/LNET.2024.3522966.

[21] G. J. W. Kathrine, P. Krittikka, I. Johnraja, S. Kirubakaran, S. Salaja, and K. Arunkumar, "Enhancing Smart Grid Security with XMSS-Based Blockchain Technology," *1st Int. Conf. Emerg. Res. Comput. Sci. ICERCS 2023 - Proc.*, 2023, doi: 10.1109/ICERCS57948.2023.10433959.

[22] Jonathan Lopez-Valdivieso, Rene Cumplido, "Design and Implementation of Hardware-Software Architecture Based on Hashes for SPHINCS+," *ACM Trans. Reconfigurable Technol. Syst.*, vol. 17, no. 4, 2024, [Online]. Available: https://dl.acm.org/doi/10.1145/3653459

[23] Liqun Chen, Changyu Dong, "Sphinx-in-the-Head: Group Signatures from Symmetric Primitives," *Cryptol. ePrint Arch.*, 2024, [Online]. Available: https://eprint.iacr.org/2024/649

[24] L. Mu, M. Lv, S. Wang, and H. Cao, "A Hybrid Index-Based Block Construction and Retrieval Algorithm for Efficient Data Retrieval on Blockchain," *2024 4th Int. Conf. Comput. Sci. Blockchain, CCSB 2024*, pp. 487–491, 2024, doi: 10.1109/CCSB63463.2024.10735653.

[25] Lily Chen, Stephen Jordan, Yi-Kai Liu, Dustin Moody, Rene Peralta, Ray Perlner, Daniel Smith-Tone, "Report on Post-Quantum Cryptography," *Natl. Inst. Stand. Technol.*, 2016, [Online]. Available: https://nvlpubs.nist.gov/nistpubs/ir/2016/nist.ir.8105.pdf

[26] Divesh Aggarwal, Gavin K. Brennen, Troy Lee, Miklos Santha, Marco Tomamichel, "Quantum attacks on Bitcoin, and how to protect against them," *arXiv:1710.10377*, 2017, [Online]. Available: https://arxiv.org/abs/1710.10377

[27] R. Roman, R. Arjona, J. Arcenegui, and I. Baturone, "Hardware Security for eXtended Merkle Signature Scheme Using SRAM-based PUFs and TRNGs," *Proc. Int. Conf. Microelectron. ICM*, vol. 2020-December, Dec. 2020, doi: 10.1109/ICM50269.2020.9331821.

[28] "A Performance Comparison of Post-Quantum Algorithms in Blockchain." Accessed: Mar. 31, 2026. [Online]. Available: https://www.researchgate.net/publication/367589436_A_Performance_Comparison_of_Post-Quantum_Algorithms_in_Blockchain

[29] "(PDF) Developing Hybrid Cryptographic Primitives for Quantum- Resistant Blockchain Technologies." Accessed: Mar. 31, 2026. [Online]. Available: https://www.researchgate.net/publication/392082360_Developing_Hybrid_Cryptographic_Primitives_for_Quantum-_Resistant_Blockchain_Technologies

[30] M. Ranjani, S. Pandey, A. Gattani, A. Choudhary, M. Gupta, and G. K. Sandhia, "Quantum Attacks on Blockchain and Mitigation Strategies," *Proc. 8th Int. Conf. Trends Electron. Informatics, ICOEI 2025*, pp. 679–685, 2025, doi: 10.1109/ICOEI65986.2025.11013287.

[31] S. K. L. P. V. Abha Naik, Esra Yeniaras, Gerhard Hellstern, Grishma Prasad, "From Portfolio Optimization to Quantum Blockchain and Security: A Systematic Review of Quantum Computing in Finance," *arXiv:2307.01155*, 2023, [Online]. Available: https://arxiv.org/abs/2307.01155

[32] W. Yang, X. Dai, J. Xiao and H. Jin, "LDV: A Lightweight DAG-Based Blockchain for Vehicular Social Networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, 2020, doi: 10.1109/TVT.2020.2963906.

[33] Longbo Han, Gengran Hu, "A Novel Lattice-Based Blockchain Infrastructure and Its Application on Trusted Data Management," *IEEE Trans. Netw. Sci. Eng.*, vol. 12, no. 4, 2025, [Online]. Available: https://ieeexplore.ieee.org/document/10922084