IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

A HYBRID POST-QUANTUM ADAPTATION FRAMEWORK FOR SECURE BLOCKCHAIN PROTOCOLS USING RSA & CRYSTALS-DILITHIUM

Srihasam Sri Lalitha[1] M.Tech[1]

Dr B. Suman[2] Associate Professor[2]

Dr V. Anantha Krishna[3] Professor[3]

Dr U. Srilakshmi[4] Professor & HOD[4]

Department of Computer Science and Engineering[1234]

Sridevi Women's Engineering College, Hyderabad, Telangana, India[1234]

Abstract – Quantum Computing represents a significant challenge to blockchain security, potentially undermining the cryptographic mechanisms that secure transactions and keep data in blockchain consistent. The main approaches used in conventional blockchain systems are based on RSA and Elliptic Curve Cryptography (ECC), which can be attacked by quantum computers. Post-Quantum Cryptography (PQC) has since become a necessity for the development of blockchain technologies that can withstand the future. This research proposes and investigates a hybrid cryptographic scheme between classical RSA and the CRYSTALS-Dilithium digital signature scheme that would enable a smooth transition to quantum-resistant blockchain systems, without disrupting the existing infrastructure.

We created a prototype implementation of the blockchain with Python, with three different cryptographic configurations: Traditional RSA/ECC, Pure CRYSTALS-Dilithium and Hybrid RSA–Dilithium. Multiple performance metrics such as average transaction latency, stable throughput, block generation time, transaction validation latency, consensus performance, CPU usage, memory usage, validator statistics, and cryptographic key and signature size were visualised with a web-based benchmarking dashboard that was built using Flask. The experimental studies were done in a synthetic blockchain network with operating conditions identical to those of the network with 2000 transactions and 5 blocks.

The results exhibit that the suggested hybrid technique yields an average transaction latency of 23.09ms and a steady throughput of 16.7 transactions per second, whereas the traditional technique has a transaction latency of 26.00ms and a throughput of 15.1 transactions per second. The hybrid approach is the consequence of a public key and signature that is much larger than that of the original public key and signature, but does offer cryptographic protection from future attacks based on quantum computers without too much computational complexity. An analysis of the comparative characteristics of all these different types reveals that the hybrid type is a good middle ground between performance and security, deployment and usage of new and legacy blockchain systems, making it a viable option for migrating.

Keywords: Blockchain Security, Post-Quantum Cryptography, CRYSTALS-Dilithium, Hybrid Cryptography, RSA, Benchmarking, Flask.

I. Introduction

Blockchain technology offers secure, transparent and decentralized transaction management for applications like finance, healthcare and supply chain management. A common authentication system and digital signature system in most blockchain systems uses RSA or Elliptic Curve Cryptography (ECC).

With the advancement of quantum computing, these traditional cryptographic algorithms may become vulnerable. We are also in the process of developing a safe solution to quantum attacks – Post-Quantum Cryptography (PQC), with CRYSTALS-Dilithium being the latest option. However, this is not easily achievable in the blockchain as it is incompatible with and slow to PQC.

To achieve quantum-resistant security while also ensuring compatibility with existing systems, a hybrid blockchain framework is proposed, which combines the security of both the RSA and CRYSTALS-Dilithium schemes. The performance of traditional RSA/ECC, CRYSTALS-Dilithium and hybrid RSA–Dilithium is compared on a Flask-based dashboard, both locally and in the cloud.

1.1 Background

The primary source of protection for blockchain is cryptographic algorithms like ECC and RSA. These algorithms are safe from classical computers but not quantum computers. CRYSTALS-Dilithium is a NIST-standardized post-quantum digital signature algorithm with greater security.

1.2 Motivation

The current blockchain technologies cannot handle quantum computers. A Hybrid strategy can be used to adopt post-quantum cryptography at a slower pace while keeping the current systems intact. Another integral benchmarking platform is required for comparison of various cryptographic approaches.

1.3 Problem Statement

The existing blockchain systems are based on RSA and ECC, both of which can be susceptible to quantum attacks. It is hard to replace them with post-quantum cryptography because of the larger key sizes, high computational requirements, and compatibility problems. Hence, a hybrid blockchain framework requiring benchmarking support is required.

1.4 Research Objectives

The aims of this work are:

- Learn how quantum computing affects the security of blockchain technology.
- Design a hybrid blockchain system based on RSA and CRYSTALS-Dilithium.
- Compare and contrast the Traditional, Dilithium and Hybrid cryptographic approaches.
- Create a benchmarking dashboard with Flask.
- Compare and measure performance based on latency, throughput, block time, CPU usage, memory usage, and cryptographic overhead.
- It can be deployed locally or on the cloud.

1.5 Research Contributions

This research has made the following major contributions:

- Design of a hybrid framework of the blockchain RSA–CRYSTALS-Dilithium.
- Implementation of Traditional, Dilithium and Hybrid cryptographic modes.
- Benchmarking dashboard for performance analysis, based on Flask.
- Support for local and Render cloud deployments.
- Multiple blockchain metrics used to compare performance.

II. Literature Review

Blockchain provides secure and decentralized transaction management using cryptographic algorithms and consensus mechanisms. But the emergence of quantum computers has sparked some worries about the security of current blockchain systems. As a potential solution to the security concerns of future blockchains, researchers have proposed post-quantum cryptography and hybrid cryptographic methods.

2.1 Blockchain Security

Security on the blockchain is ensured by various means, including cryptographic algorithms, digital signatures, consensus mechanisms, and decentralised data storage. These techniques guarantee that transactions are authentic and secure. But existing cryptographic methods could be broken by future quantum attacks.

2.2 Classical Cryptography in Blockchain

RSA and Elliptic Curve Cryptography (ECC) are commonly employed in most blockchain systems for authentication and digital signatures. These algorithms are safe from classical computers and could be broken by quantum algorithms like Shor's.

2.3 Post-Quantum Cryptography

Post-Quantum Cryptography (PQC) offers cryptographic algorithms secure against both classical and quantum attacks. NIST has standardized a number of PQC algorithms for future use. PQC provides more security but has larger keys and needs more computational power.

2.4 CRYSTALS-Dilithium

CRYSTALS-Dilithium is a NIST standardized Lattice-based Digital Signature Algorithm. It offers good quantum resistance security, with more efficient signature generation and verification, but creates bigger keys and signatures than the traditional methods.

2.5 Hybrid Cryptographic Approaches

The idea is to use a hybrid cryptography scheme that merges classic algorithms like RSA with post-quantum algorithms like CRYSTALS-Dilithium. This strategy allows for better security without any need to break away from the existing blockchain systems, allowing for a step-by-step transition to quantum-resistant cryptography.

2.6 Research Gap

Most of the existing blockchain systems are still based on RSA/ECC, which are susceptible to future quantum attacks. CRYSTALS-Dilithium offers “quantum-resistant” security, but there is not much research that integrates classical and post-quantum cryptography into one blockchain system. There are also no systems currently in place that provide a comprehensive platform for benchmarking solutions.

To overcome these problems, this paper introduces a hybrid blockchain framework that combines the advantages of the RSA blockchain and the Dilithium blockchain, and presents a benchmarking dashboard based on Flask that can be executed locally and deployed to the cloud through Render.

III. Existing System

Existing blockchain systems mainly use RSA and Elliptic Curve Cryptography (ECC) for digital signatures and transaction authentication. These systems offer secure and distributed transaction processing, but are susceptible to future quantum attacks. Most of the current solutions also don't support postquantum nor provide a full range of benchmarking.

3.1 Overview of Existing Blockchain Security

The security of blockchain is rooted in its cryptographic algorithms, consensus protocols, and decentralized network. Transactions are digitally signed and verified before being added to the blockchain. But the basic cryptographic techniques are not necessarily secure on quantum computers.

3.2 RSA/ECC-based Blockchain

Digital signatures and authentication are commonly implemented with RSA and ECC. They are robust against classical attacks, but are susceptible to quantum algorithms like Shor's algorithm.

3.3 Existing Benchmarking Approaches

These are the metrics that are currently being measured and benchmarked as metrics for existing tools: Latency, Throughput, and block generation time. But most tools only consider traditional cryptographic approaches and not classical, post-quantum and hybrid approaches.

3.4 Limitations of Existing Systems

The current blockchain systems have the following drawbacks:

- **Quantum Vulnerability:** RSA and ECC are susceptible to quantum attacks.
- **Limited Migration Support:** There is no smooth migration to post-quantum cryptography for the existing systems.
- **No Benchmarking:** Not many platforms compare traditional, post-quantum and hybrid cryptography.
- **No Hybrid Framework:** Existing systems normally only have a single cryptographic approach supported.

IV. Proposed System

The system being proposed includes a hybrid approach between RSA and CRYSTALS-Dilithium, which would enable a smooth and secure migration to post-quantum blockchain security. It works in three modes: Traditional RSA/ECC, CRYSTALS-Dilithium, and Hybrid RSA–Dilithium. The performance of the blockchain is benchmarked using a Flask-based dashboard in both local and Render cloud environments.

4.1 Proposed Framework

The framework consists of blockchain modules, cryptographic modules, a benchmark engine and a Flask dashboard. It enables users to run transactions, compare various cryptographic schemes, and analyze the performance of the schemes.

4.4.1 Execution Modes

In the proposed framework, there are two ways of executing.

- **Local Execution**: Executes the Application locally with Python and Flask.
- **Cloud Deployment (Render):** Hosts the application in the cloud using the Render cloud platform.

4.2 System Architecture

The framework comprises the User Interface, Blockchain Core, RSA, Dilithium, Hybrid Cryptography, Consensus Module, Benchmark Engine, and Analytics Dashboard. The transactions are processed, verified, recorded on the blockchain, and the results of the benchmarks are shown in the dashboard.

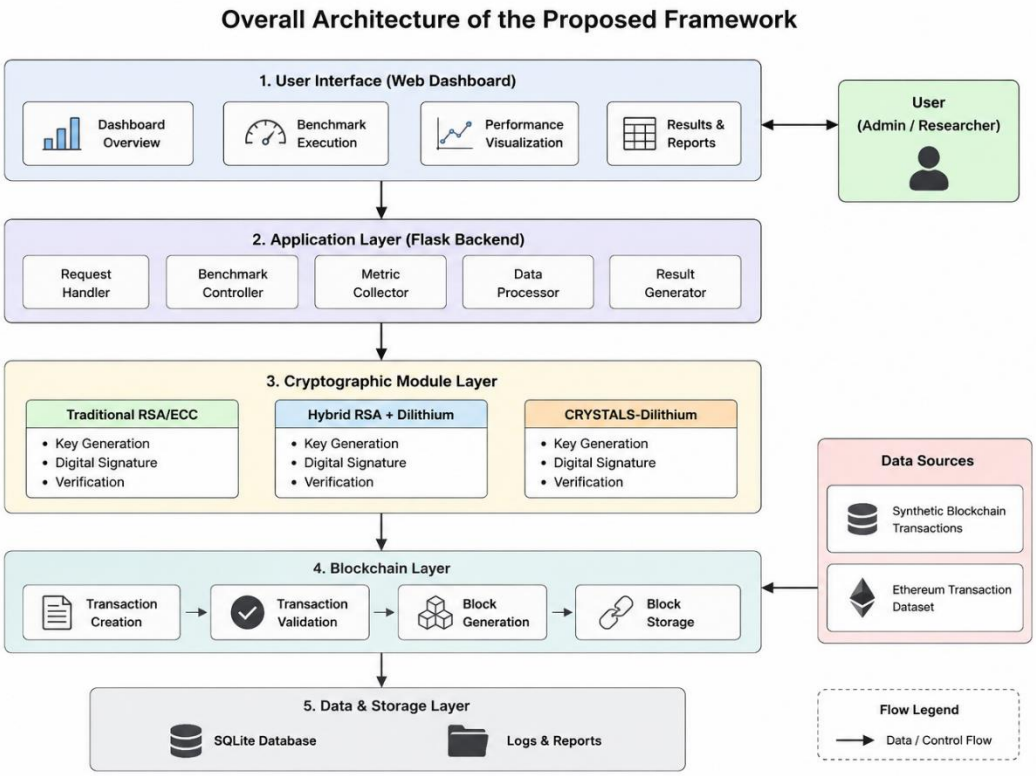

Figure 4.1: *System Architecture of the proposed framework*

4.3 Design Modules

The proposed framework comprises the following modules:

- **User Interface:** Runs benchmark experiments.
- **Blockchain Core:** Manages blockchain operations.
- **RSA Module:** Performs RSA signing and verification.
- **Dilithium Module:** Implements post-quantum signatures.
- **Hybrid Module:** Combines RSA and Dilithium.

- **Transaction Processing:** Handles transaction execution.
- **Consensus Module:** Validates transactions.
- **Benchmark Engine:** Measures performance metrics.
- **Analytics Dashboard:** Displays benchmark results.

4.4 Workflow of the Proposed System

The user chooses a cryptographic mode, runs a benchmark, and the transaction is signed, verified and added to the blockchain. The dashboard collects and displays the performance metrics.

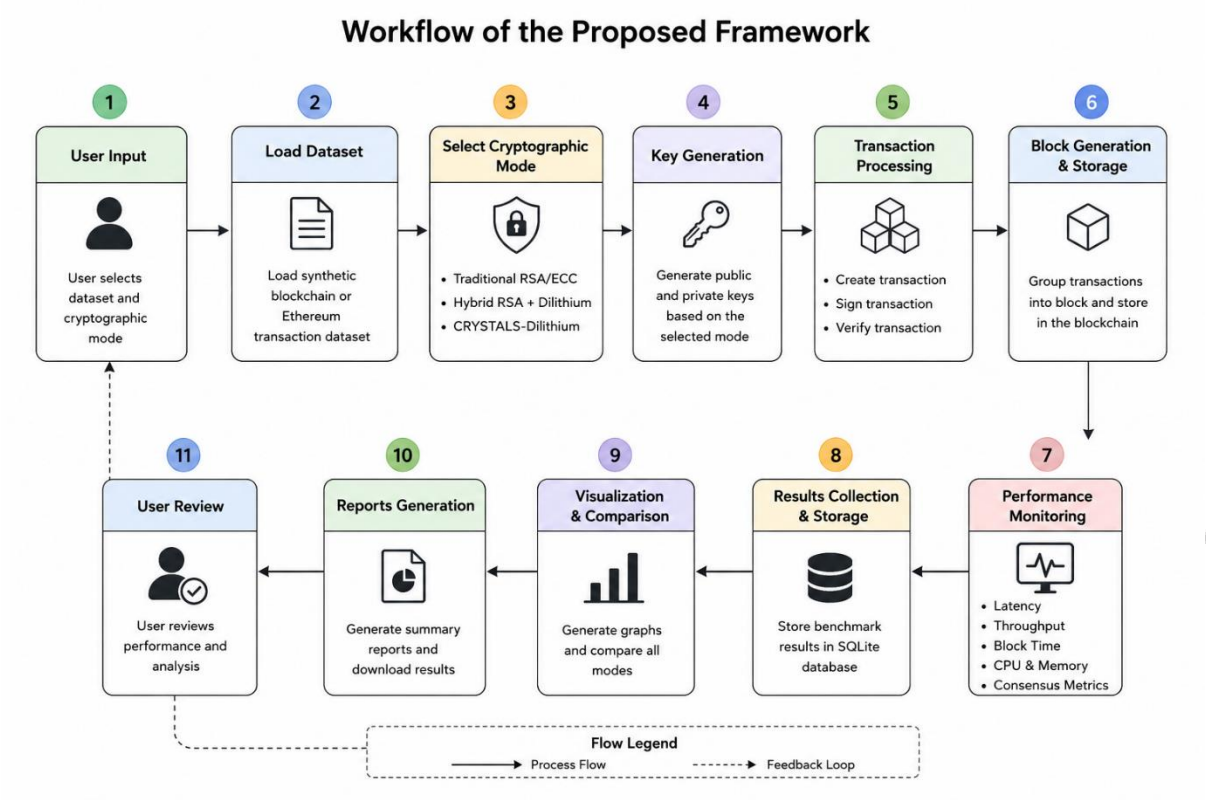

Figure 4.2: *Workflow of the Proposed Framework*

4.5 Proposed Transaction Signing Algorithm

The proposed transaction signing process creates digital signatures for the given cryptographic configuration.

Algorithm 1: Proposed Transaction Signing Algorithm

Input: Data for Transaction, Selected Cryptographic Mode

Output: Digitally Signed Transaction

- Get transaction information from the user.
- Choose between the cryptographic modes (Traditional, Dilithium, or Hybrid).
- Generate public and private keys.
- Generate a digital signature of the transaction.
- Attach the signature to the transaction.

- Send the signed transaction for verification.

4.6 Proposed Hybrid Verification Algorithm

When hybrid is chosen, the hybrid verification algorithm is used to validate transactions, making use of both RSA and CRYSTALS-Dilithium.

Algorithm 2: Proposed Hybrid Verification Algorithm

Input: Signed Transaction

Output: Verified Transaction

- Take the signed transaction.
- Specify the mode of the selected cryptographic algorithm.
- Check RSA signature (if applicable).
- Ensure correct Dilithium signature (if applicable).
- Approve the transaction if all necessary signatures are good.
- Add the verified transaction to the blockchain.
- Otherwise, reject the transaction.

4.7 Security Model

It includes RSA compatibility, along with CRYSTALS-Dilithium quantum resistance. It provides authentication of transactions in a secure manner, and it allows for performing one or more transactions and comparing the performance of various cryptographic methods.

4.8 Advantages of the Proposed Framework

The suggested system has the following benefits:

- Quantum-resistant security. Supports Traditional, Dilithium and Hybrid modes.
- Available with existing blockchain systems.
- Flask-based benchmarking dashboard.
- Enables local or cloud execution.
- Compares several performance measures.
- Supports “post-quantum” blockchain security with hands-on transition.

V. Implementation Framework

The proposed hybrid blockchain framework was created with the Python programming language and Flask. It applies to three cryptographic modes: Traditional RSA/ECC, CRYSTALS-Dilithium and Hybrid RSA–Dilithium. It can be run locally or deployed to the Render cloud platform.

5.1 Development Environment

It was implemented in Python, using Flask for the web application. HTML, CSS, and JavaScript were used for the user interface, while SQLite was used to store blockchain and benchmark data. The app was tested locally and then deployed to Render.

5.2 Software Requirements

The software requirements needed are as follows:

- **Operating System:** Windows
- **Programming Language:** Python 3.x
- **Web Framework:** Flask
- **Database:** SQLite
- **Front-end:** HTML, CSS, JavaScript
- **Visualization Library:** Chart.js
- **Web Browser:** Google Chrome
- **Cloud Platform:** Render

5.3 Hardware Requirements

The framework is not based on high-end computers and may be executed on a standard PC.

- **Processor:** Intel Core i5 or equivalent
- **RAM:** 8 GB or higher
- **Storage:** 10 GB available disk space
- **Internet Connection:** Required for cloud deployment

5.4 Technology Stack

The proposed framework uses the following technologies:

- **Python:** Python provides the implementation of blockchain logic and cryptographic operations.
- **Flask:** Works on building the Internet-based benchmarking application.
- **HTML/CSS:** Designs the user interface.
- **JavaScript:** Performs actions on the web page.
- **Chart.js:** Shows benchmark results in a graph.
- **SQLite:** Contains blockchain and benchmark data.
- **CRYSTALS-Dilithium:** Digital signatures based on a post-quantum cryptographic method.
- **RSA:** Provides traditional digital signatures and hybrid authentication.

5.5 Blockchain Implementation

The blockchain module handles blockchain transaction operations, digital signing, verification, creation and management. It allows you to use both synthetic and Ethereum transaction datasets and supports Traditional, Dilithium and Hybrid cryptographic modes.

5.6 Dashboard Implementation

The dashboard is built using Flask and offers the user the option to choose cryptographic modes, execute benchmarks and present results on summary cards, tables and graphs.

5.7 Benchmark Engine Implementation

These experiments and measurements are automatically performed by the benchmark engine:

- Average transaction latency
- Stable throughput
- Block generation time
- Transaction validation latency
- CPU usage
- Memory usage
- Consensus performance
- Public key size
- Signature size
- Validator statistics

This outcome is shown on the dashboard and compared.

5.8 Local Execution Implementation

The framework can be implemented locally with Python. The Flask dashboard can be accessed using a web browser to run benchmarks and view results.

5.9 Cloud Deployment using Render

The framework is deployed on the Render platform so that users can view the benchmarking dashboard directly in the browser, without having to install anything. It offers the same features as the local version for use in remote tests and performance evaluation.

VI. Experimental Setup

6.1 Experimental Environment

The proposed framework was tested by implementing the Python-based blockchain application in the Flask dashboard. Two datasets were used for experiments, namely:

- Synthetic blockchain transaction dataset
- Ethereum transaction dataset

All experiments had the same blockchain setup for a fair comparison.

6.2 Blockchain Configuration

Fixed parameters of the blockchain were set as follows:

- **Number of Validators:** Fixed throughout all experiments
- **Number of Transactions:** 2000
- **Block Size:** Fixed block size
- **Mining Parameters:** Same for all cryptographic configurations

6.3 Cryptographic Configurations

The framework has three cryptographic modes:

- **RSA-2048:** Traditional digital signatures
- **CRYSTALS-Dilithium:** Post-quantum digital signatures
- **Hybrid RSA + Dilithium:** Combination of RSA and Dilithium

6.4 Performance Metrics

The benchmark engine has the following dimensions:

- Average Latency
- Stable Throughput
- Block Time
- CPU Usage
- Memory Usage
- Consensus Performance
- Public Key Size
- Signature Size
- Validator Statistics

6.5 Experimental Platform

Both models were successfully implemented in accordance with the plan:

- Local Python environment
- Render cloud platform

The scores of the benchmark were comparable in both environments.

VII. Results and Discussion

The proposed framework is tested with the same experimental settings as the following: Traditional RSA/ECC, CRYSTALS-Dilithium and Hybrid RSA-Dilithium. The benchmark results were taken via the Flask dashboard and evaluated with varying performance metrics.

7.1 Traditional RSA Results

The Traditional RSA/ECC implementation was used as the baseline.

Synthetic Dataset

- Average Latency: **26.00 ms**
- Throughput: **15.1 tx/s**
- Block Generation Time: **645.67 ms**

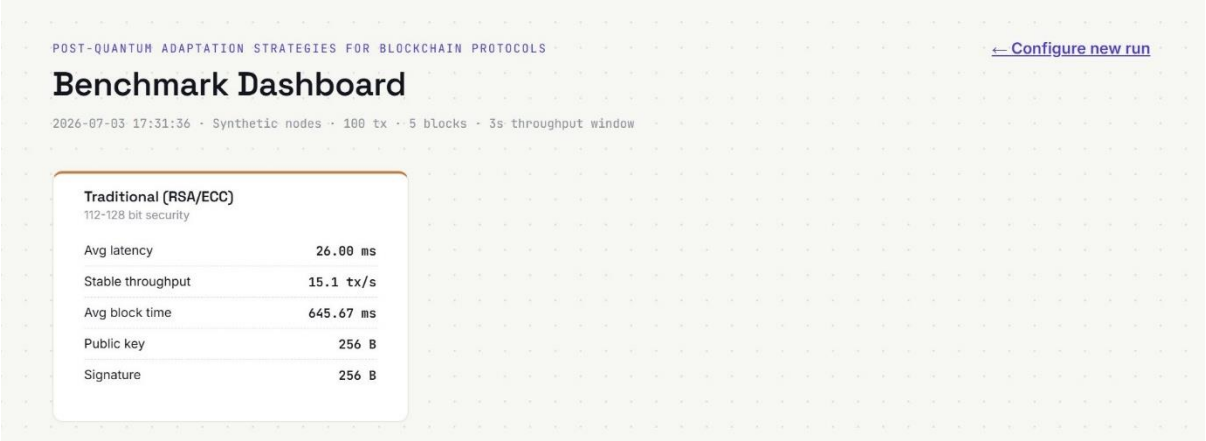

Figure 7.1: *Traditional RSA Benchmark Summary (Synthetic Dataset)*

Ethereum Dataset

- Average Latency: **31.77 ms**
- Throughput: **15.1 tx/s**
- Block Generation Time: **696.04 ms**

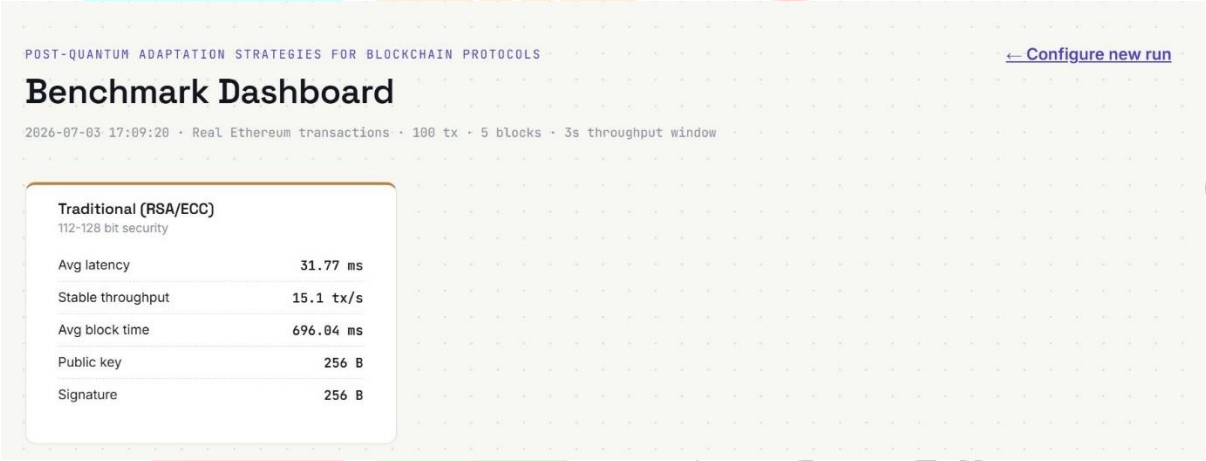

Figure 7.2: *Traditional RSA Benchmark Summary (Ethereum Dataset)*

7.2 CRYSTALS-Dilithium Results

Synthetic Dataset

- Average Latency: **22.16 ms**
- Throughput: **18.8 tx/s**
- Block Generation Time: **593.39 ms**

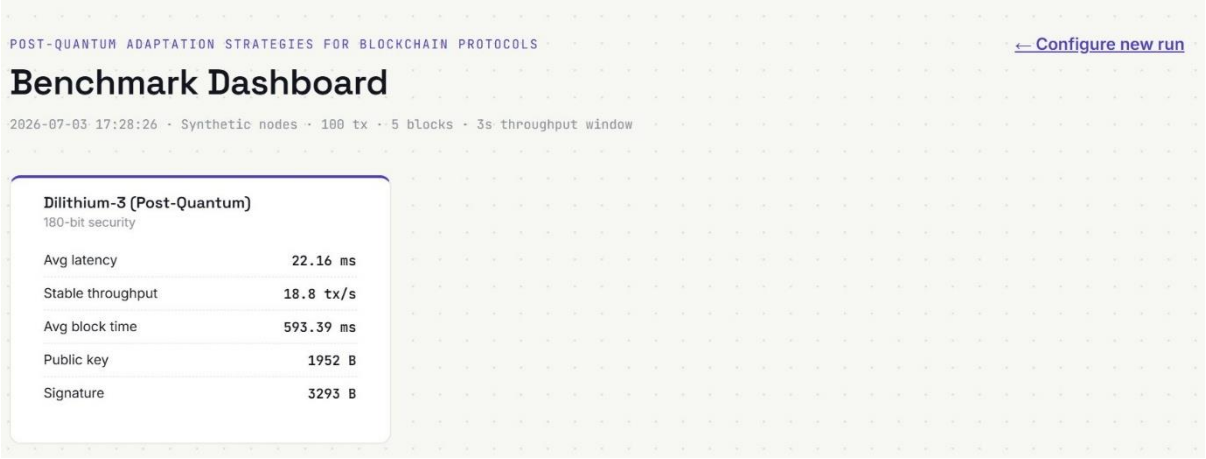

Figure 7.3: *CRYSTALS-Dilithium Benchmark Summary (Synthetic Dataset)*

Ethereum Dataset

- Average Latency: **20.10 ms**
- Throughput: **19.4 tx/s**
- Block Generation Time: **660.49 ms**

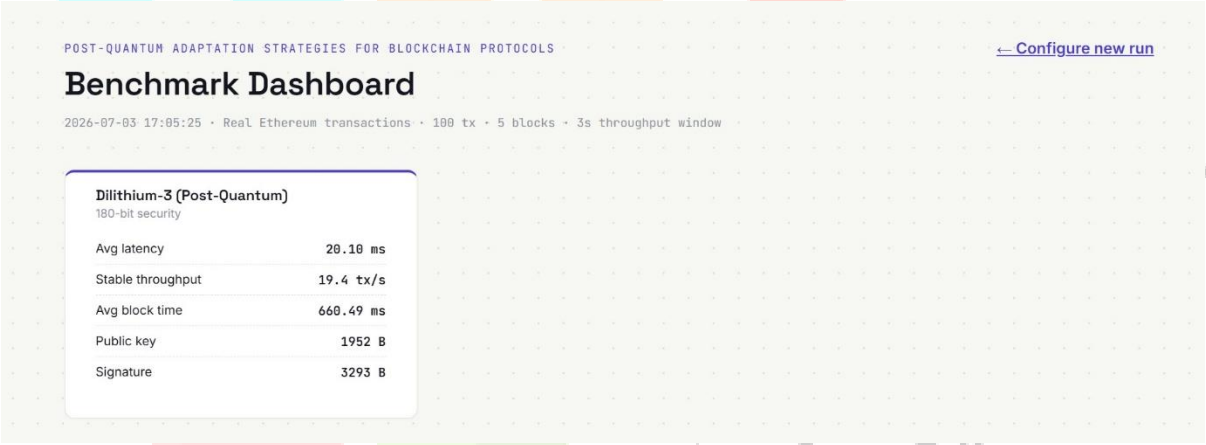

Figure 7.4: *CRYSTALS-Dilithium Benchmark Summary (Ethereum Dataset)*

7.3 Hybrid RSA + Dilithium Results

Synthetic Dataset

- Average Latency: **23.09 ms**
- Throughput: **16.7 tx/s**
- Block Generation Time: **688.84 ms**

Figure 7.5: *Hybrid RSA + Dilithium Benchmark Summary (Synthetic Dataset)*

Ethereum Dataset

- Average Latency: **22.14 ms**
- Throughput: **18.5 tx/s**
- Block Generation Time: **619.92 ms**

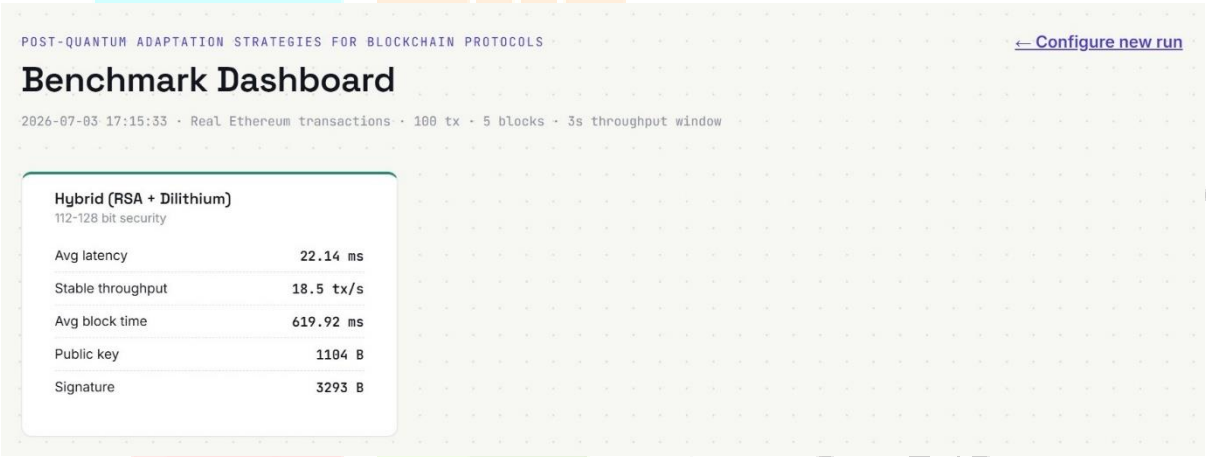

Figure 7.6: *Hybrid RSA + Dilithium Benchmark Summary (Ethereum Dataset)*

7.4 Comparative Performance Analysis

A comparative study is conducted to evaluate the three different configurations of cryptographic methods using the same configuration of blockchain.

Table 7.1: *Comparison Using Ethereum Transaction Dataset*

Metric	Traditional	Hybrid	Dilithium
Average Latency (ms)	31.77	22.14	**20.10**
Throughput (tx/s)	15.1	18.5	**19.4**
Block Time (ms)	696.04	**619.92**	660.49
Public Key (Bytes)	**256**	1104	1952
Signature Size (Bytes)	**256**	3293	3293

Table 7.2: Comparison Using Synthetic Blockchain Dataset

Metric	Traditional	Hybrid	Dilithium
Average Latency (ms)	26.00	23.09	**22.16**
Throughput (tx/s)	15.1	16.7	**18.8**
Block Time (ms)	645.67	688.84	**593.39**
Public Key (Bytes)	**256**	1104	1952
Signature Size (Bytes)	**256**	3293	3293

7.5 Dashboard Visualization

Flask' offers an interactive dashboard for comparing the performance of blockchain. Users can choose to use any one of the datasets, configure the set of cryptographic operations, run the benchmarks, and display the results on summary cards, tables, and graphs.

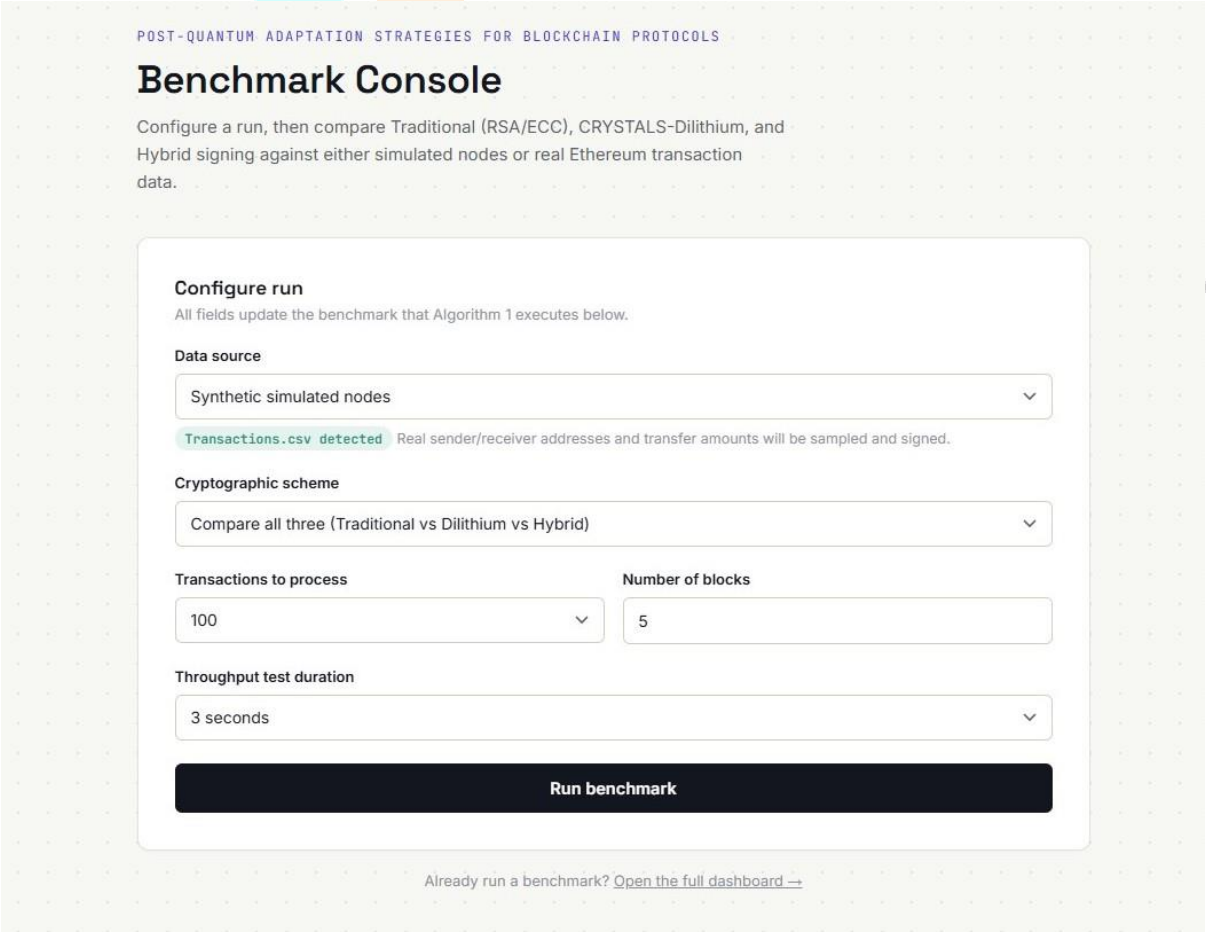

Figure 7.7: Benchmark Console (Home Page)

7.6 Performance Analysis

The benchmark results show that:

- CRYSTALS-Dilithium had the smallest transaction latency.
- Performance of hybrid RSA–Dilithium was better than Traditional RSA/ECC.
- The block generation time did not change for all the configurations.
- Hybrid and Dilithium needed bigger keys and signatures and offered more quantum-resistant security.

- The CPU and memory usage was a little high but still acceptable.
- The results of consensus performance were similar for each of the three methods.
- In general, the hybrid implementation offers a decent level of security, performance and compatibility.

7.7 Benchmark Summary

The proposed framework can successfully compare the three systems Traditional RSA/ECC, CRYSTALS-Dilithium and Hybrid RSA–Dilithium under identical conditions. The Hybrid approach is a reasonable migration path that offers a balance of compatibility, security, and performance, while CRYSTALS-Dilithium provides the best performance and quantum-resistant security.

7.8 Performance Graphs

The benchmarking dashboard automatically creates graphs of:

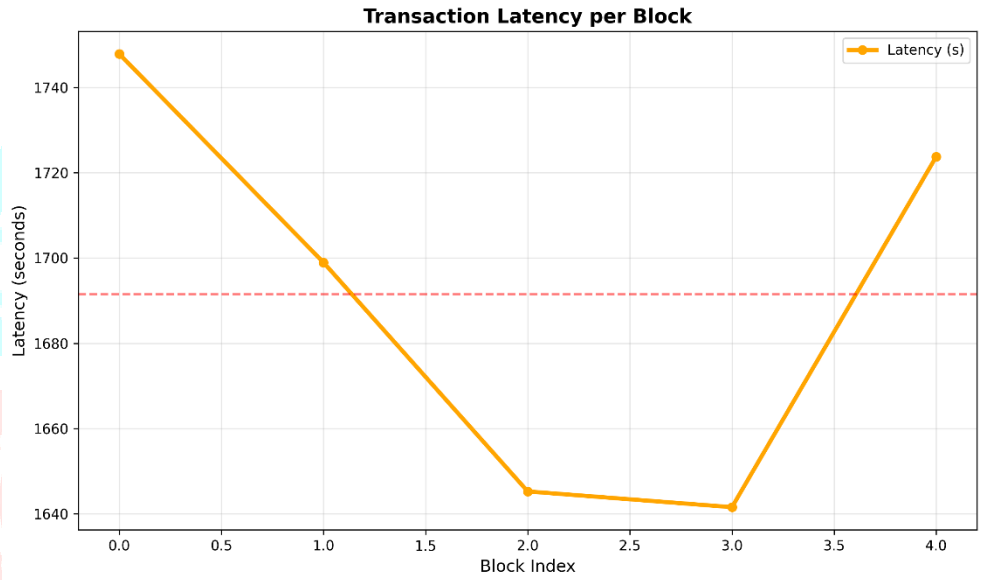

Figure 7.8: *Block-wise Transaction Latency*

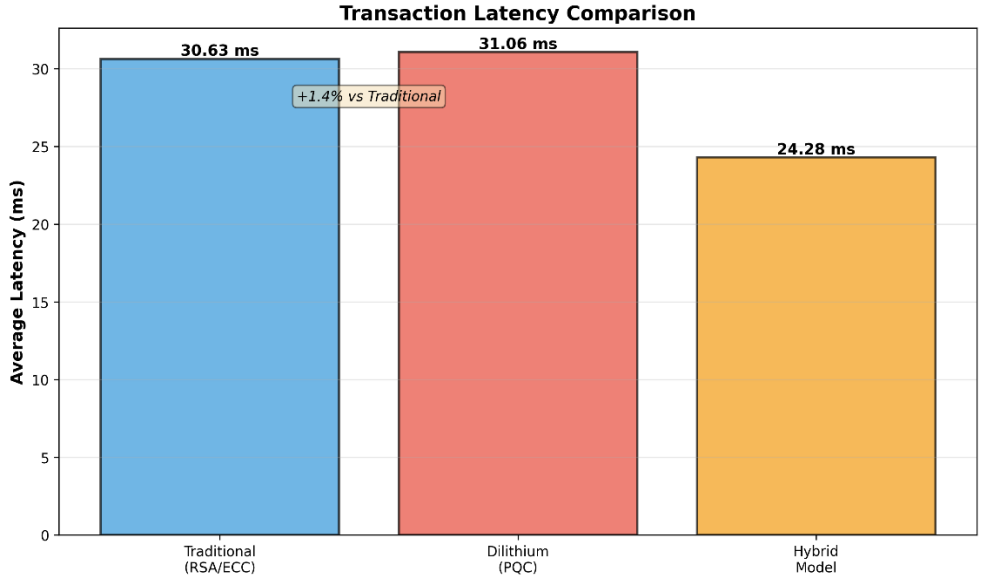

Figure 7.9: *Average Transaction Latency Comparison*

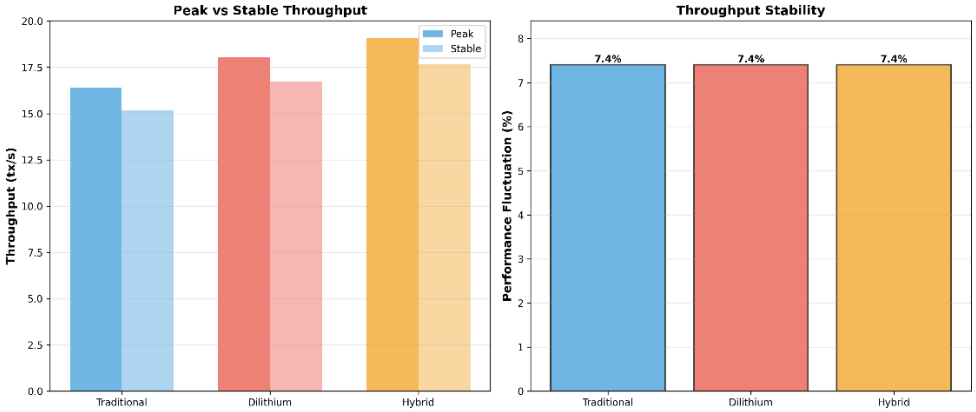

Figure 7.10: *Peak and Stable Throughput Comparison*

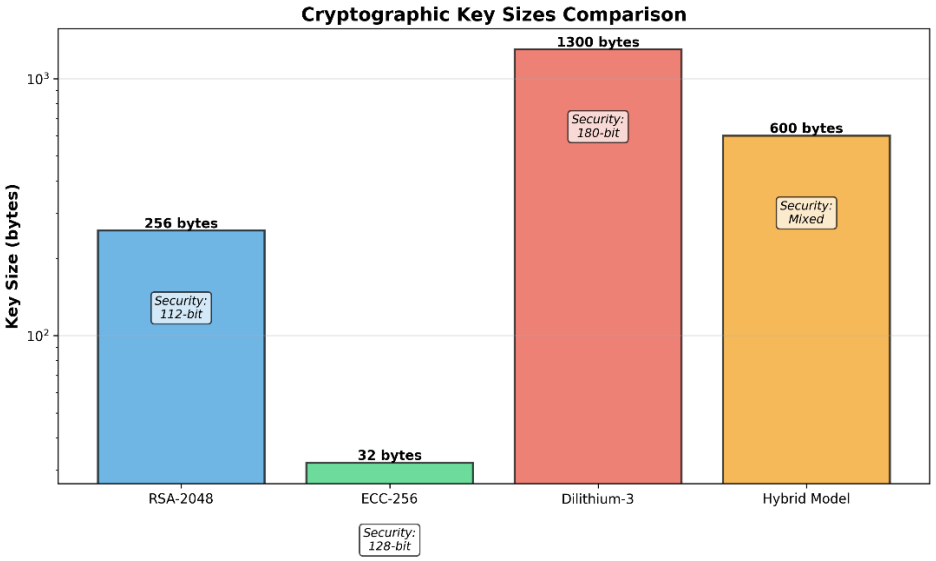

Figure 7.11: *Cryptographic Key Size Comparison*

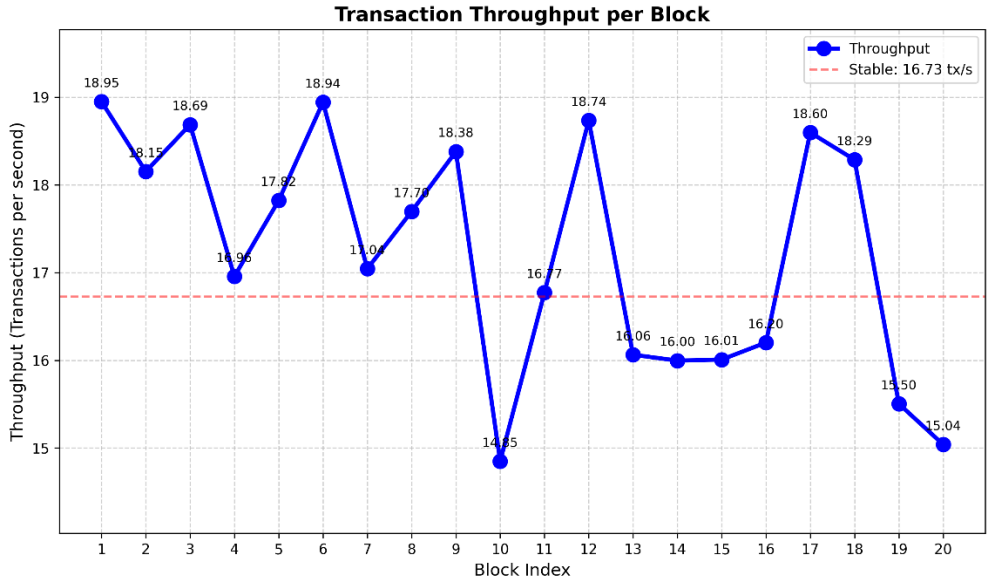

Figure 7.12: *Transaction Throughput over Time*

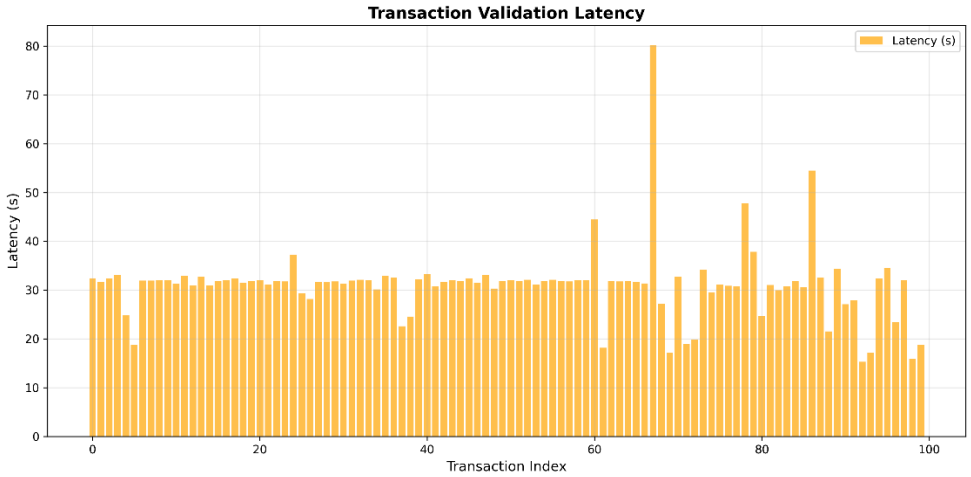

Figure 7.13: *Transaction Validation Latency*

These graphs offer a straightforward comparison of the three configurations used for cryptography and allow for the performance and security characteristics of each configuration to be analysed.

VIII. Advantages and Limitations

This proposed hybrid blockchain is a combination of classical and post-quantum cryptography, which improves the security of the blockchain. It is more efficient and enables transition to quantum-resistant blockchain systems. There are, however, some disadvantages to it.

8.1 Advantages of the Proposed System

The proposed framework has the following benefits:

- **Quantum-resistant security:** Based on CRYSTALS-Dilithium, which is quantum-resistant.
- **Improved throughput:** Greater number of transactions than the Traditional RSA/ECC approach.
- **Lower transaction latency:** Reduces transaction processing time.
- **Backward compatibility:** Supports both traditional and post-quantum cryptography.
- **Smooth migration:** Allows for progressive implementation of post-quantum security.
- **Interactive benchmarking dashboard:** Provides real-time performance visualization.
- **Modular design:** Allows the framework to be more easily maintained and extended.
- **Scalable framework:** Supports both local execution and cloud deployment using Render.

8.2 Limitations

The proposed framework has the following limitations:

- **Larger key and signature sizes:** Hybrid and Dilithium require more storage than Traditional RSA/ECC, due to larger key and signature sizes.
- **Higher computational overhead:** Additional processing is needed for post-quantum cryptography, causing greater computational overhead.
- **Synthetic benchmarking:** Synthetic transaction datasets are used in some experiments.
- **Limited validator network:** A limited number of validators were used for testing.

- **No public blockchain deployment:** The framework was only tested in local environments and in the cloud.
- **Additional storage overhead:** Keys and signatures are larger, adding to storage requirements.

Overall, the proposed framework provides a practical approach for improving blockchain security while supporting a smooth transition to post-quantum cryptography.

IX. Conclusion and Future Work

9.1 Conclusion

A hybrid post-quantum blockchain framework that combines Traditional RSA/ECC and CRYSTALS-Dilithium was proposed in this paper to strengthen the security of the blockchain system in the quantum era. The three setups of cryptography are Traditional RSA/ECC, CRYSTALS-Dilithium and Hybrid RSA–Dilithium.

A benchmarking dashboard based on Flask was created for testing the performance of the blockchain with various metrics like latency, throughput, block generation time, CPU usage, memory usage, consensus performance, key size, and signature size. This was successfully tested with local and Render cloud environments.

Based on the experimental results, the Hybrid Approach and CRYSTALS-Dilithium Approach are more secure and better performing than the Traditional Approach of RSA/ECC. Overall, the proposed framework offers a viable way to make the shift to post-quantum security for current blockchain systems.

9.2 Future Work

The proposed framework can be further extended in various directions for the improvement of the functionality and applicability.

- Deploy the framework in a real blockchain environment to validate the framework in various use cases.
- Develop other post-quantum algorithms such as Falcon and SPHINCS+.
- Do extensive benchmarking with bigger blockchain networks and larger volumes of transactions.
- Integrate smart contracts into decentralized applications.
- Implement inter-chaining between blockchains in the framework.
- Discuss the use of Hardware Acceleration to improve the performance of crypto.
- Enhance cloud deployment for scalable distributed benchmarking and remote access.

This enhancement will continue to enhance the scalability, security and user-friendliness of post-quantum blockchain networks.

X. References

[1] S. Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System*, 2008.

[2] S. Haber and W. S. Stornetta, "How to Time-Stamp a Digital Document," *Journal of Cryptology*, vol. 3, no. 2, pp. 99–111, 1991.

[3] W. Diffie and M. E. Hellman, "New Directions in Cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, Nov. 1976.

[4] R. L. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[5] V. S. Miller, "Use of Elliptic Curves in Cryptography," in *Advances in Cryptology – CRYPTO '85*, 1986, pp. 417–426.

[6] P. W. Shor, "Algorithms for Quantum Computation: Discrete Logarithms and Factoring," in *Proc. 35th Annual Symposium on Foundations of Computer Science*, 1994, pp. 124–134.

[7] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press, 2010.

[8] National Institute of Standards and Technology, *Post-Quantum Cryptography Standardization Project*.

[9] National Institute of Standards and Technology, *FIPS 204: Module-Lattice-Based Digital Signature Standard (ML-DSA)*, Aug. 2024.

[10] L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehlé, "CRYSTALS-Dilithium: A Lattice-Based Digital Signature Scheme," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2018.

[11] D. J. Bernstein and T. Lange, *Post-Quantum Cryptography*. Springer, 2009.

[12] D. Stebila and M. Mosca, "Post-Quantum Key Exchange for the Internet and the Open Quantum Safe Project," *Lecture Notes in Computer Science*, Springer.

[13] X. Xu, I. Weber, and M. Staples, *Architecture for Blockchain Applications*. Springer, 2019.

[14] M. Swan, *Blockchain: Blueprint for a New Economy*. O'Reilly Media, 2015.

[15] A. M. Antonopoulos, *Mastering Bitcoin*, 2nd ed. O'Reilly Media, 2017.

[16] M. Crosby, P. Pattanayak, S. Verma, and V. Kalyanaraman, "Blockchain Technology: Beyond Bitcoin," *Applied Innovation Review*, vol. 2, 2016.

[17] National Institute of Standards and Technology, *Blockchain Technology Overview (NISTIR 8202)*, 2018.

[18] Ethereum Foundation, *Ethereum Documentation*.

[19] Python Software Foundation, *Python Documentation*.

[20] Flask Documentation.

[21] Chart.js Documentation.

[22] SQLite Documentation.

[23] Render Documentation.

[24] "Blockchain Security: A Survey of Techniques and Challenges," *IEEE Access*, 2021.

[25] "Post-Quantum Cryptography for Blockchain Systems: A Survey," *Future Generation Computer Systems*, 2022.

[26] "Lattice-Based Cryptography for Blockchain Applications," *Journal of Information Security and Applications*, 2022.

[27] "Hybrid Cryptographic Frameworks for Quantum-Resistant Blockchain," *IEEE Access*, 2023.

[28] "Blockchain Benchmarking: Metrics, Challenges and Evaluation," *Future Internet*, 2022.

[29] "Performance Evaluation of Blockchain Systems Using Benchmarking Frameworks," *Journal of Systems Architecture*, 2023.

[30] "Quantum Computing and Its Impact on Blockchain Security," *ACM Computing Surveys*, 2021.

[31] "Migration Strategies Toward Post-Quantum Blockchain Systems," *IEEE Access*, 2023.

[32] "Performance Analysis of CRYSTALS-Dilithium in Blockchain Applications," *Electronics*, 2023.

[33] "Hybrid Digital Signature Schemes for Secure Blockchain Networks," *Computer Standards & Interfaces*, 2024.

[34] "PQS-BFL: A Post-Quantum Secure Blockchain-based Federated Learning Framework," 2025.

[35] "Benchmarking NIST-Standardised ML-KEM and ML-DSA on ARM Cortex-M0+," 2026.