International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 10 October 2026
Page No: 5762 - 5771
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0748

QShieldChain: A Quantum-Resistant Blockchain Framework Using CRYSTALS-Dilithium3 and AES for Secure Decentralized Transactions

U. Sudheer[1], P. Shyamala Madhuri[2]
[1,2]Department of Computer Science and Engineering, Vishnu Institute of Technology, Bhimavaram, Andhra Pradesh, India.
Email Id: *24PA1D5818@vishnu.edu.in[1], shyamalamadhuri.y@vishnu.edu.in[2]*

Abstract
The security of traditional blockchain systems has been undermined by quantum computing through the following scenarios: – Traditional cryptographic algorithms like RSA and Elliptic Curve Cryptography (ECC) are vulnerable to quantum attacks. To mitigate these important vulnerabilities, this work proposes a secure and private blockchain system that utilizes two cryptographic methods: the CRYSTALS-Dilithium post-quantum digital signature and Advanced Encryption Standard (AES) symmetric encryption. The system proposed here improves the security of transactions in the system: post-quantum cryptography is used for verifying transactions, and AES encryption is used to keep the data in the transaction confidential. The proposed extension would encrypt all transaction information before storage, unlike existing blockchain systems that store the information in plain text, which would make it impossible for any unauthorized parties to access the information. It provides user registration, transaction generation, validation and storage on the blockchain via Ethereum smart contracts. The reason for the choice is that Dilithium3 is more secure, and has optimized latency. Experimental analysis shows that the proposed model provides high security, reliable transaction verification and better security from quantum computing attacks with acceptable performance. Comparative analysis shows that Dilithium3 offers a better key-size to latency tradeoff than both RSA and ECC, and has greater security guarantees. This study suggests a two-layer security model comprising AES encryption and CRYSTALS-Dilithium3 digital signatures in the context of Ethereum based blockchain environment, as a novel approach to secure decentralized transaction systems in post-quantum era. The framework is validated using a functional simulation environment that can be quantitatively evaluated.
Keywords: *Quantum-resistant cryptography, CRYSTALS-Dilithium, Blockchain security, Post-quantum cryptography, AES encryption, Ethereum smart contracts, Lattice-based cryptography, Digital signatures, Transaction validation, Decentralized systems.*

1. Introduction
Blockchain technology is a new paradigm in distributed computing, which provides secure, transparent and tamper-proof management of transactions without the need of any centralized authority [1]. Current blockchain systems rely on the well-known cryptographic tools, such as Elliptic Curve Cryptography (ECC) and RSA (Rivest–Shamir–Adleman) that implement digital signatures and key exchange for transaction integrity and user authentication [2]. But with the coming of quantum computing, these traditional cryptographic methods are in danger. The RSA problem is based on the hardness of integer factorization, and the ECC problem on the hardness of discrete log, which can both be solved in polynomial time by executable Shor algorithm on a quantum processor powerful enough [3]. With RSA or ECC, digital signatures would then be easily forgeable and the hacker would still have access to the blockchain's historical records, which are already difficult to change due to their immutability. To meet this challenge, the National Institute of Standards and Technology (NIST) Post-Quantum Cryptography Standardization Program (PQCSP) has evaluated and standardized quantum-resistant algorithms [4]. Relative to other finalists, CRYSTALS-Dilithium is a digital signature scheme based on the Module Learning with Errors (MLWE) problem with good security proofs, relatively small key sizes compared to hash-based schemes, and efficient performance. Even though there are

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 10 October 2026
Page No: 5762 - 5771
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0748

primitives available for post-quantum cryptography, it is not fully explored how to incorporate quantum-resistant cryptography into operational blockchain architectures [5]. The current blockchain platforms such as Ethereum, Bitcoin, and Hyperledger Fabric still use classical cryptography methods to sign and validate transactions, and adversaries can still harvest the data of the blockchain today and decrypt it later using quantum computers [6]. Furthermore, the current study on quantum secure blockchain solutions mostly just concentrates on the replacement of signatures, while omitting to consider the confidentiality of transactions [7, 8]. In a deployed Ethereum network, each transaction parameter (sender address, receiver address, amount and time) is stored and broadcasted in plain text, visible to any node that has the contract address. This feature of architecture is a major privacy risk, especially for financial and sensitive data applications. In this paper, both aspects of security are tackled by proposing a novel blockchain framework combining the quantum-resistant authentication of the CRYSTALS-Dilithium3 digital signature with quantum-resistant confidentiality of the transactions using AES-256 symmetric encryption. Empirical performance analysis shows that Dilithium3 is chosen over Dilithium2 and Dilithium5 due to its better latency and key size properties. The system is deployed and tested on an Ethereum blockchain testnet using Python web app and Solidity smart contract. The key features of the work are: (i) A dual-layer security design with quantum-resistant Dilithium3 signatures and AES encryption for blockchain transactions; (ii) A Solidity smart contract design that enables encrypted transaction storage and quantum-safe signature verification; (iii) Quantitative comparison of the latency and key size of the RSA, ECC, Dilithium2, and Dilithium3 algorithms; and (iv) An end-to-end simulation that demonstrates secure user registration, transaction creation, broadcasting, validation, and decryption. The next section will briefly discuss the related work on quantum-resistant blockchain protocols (Section 2). The proposed methodology and system architecture are provided in Section 3. Experimental results and performance analysis are given in Section 4. Finally, Section 5 presents a summary and summarizes future research directions.

2. Literature Survey

The convergence of quantum computing risks and blockchain security has sparked a great deal of research. In this section, some of the seminal and recent work relevant to this proposed work is reviewed. One of the first systematic studies of the risks of quantum computing to blockchain platforms was conducted by Fedorov et al. [2]. The authors proved that quantum computers, which could implement Shor's algorithm on future hardware, could break ECDSA-based signature schemes used by Bitcoin and Ethereum in under an hour, thus compelling a need for the migration to post-quantum. Bernstein [3] provided a comprehensive encyclopedia treatment of post-quantum cryptography, covering lattice-based, hash-based, code-based and multivariate polynomial schemes. The work identified a few contenders as top contenders for standardization, namely the CRYSTALS family (Dilithium for signatures, Kyber for key encapsulation), which was used as a base for future applied blockchain research. The NIST Status Report [4] on the third round of the Post-Quantum Cryptography Standardization process was released that formally recommends CRYSTALS-Dilithium (FIPS 204), CRYSTALS-Kyber (FIPS 203), and SPHINCS+ for standardization. This report laid down the groundwork for post-quantum algorithm implementation in security-critical systems, such as blockchain. Prajapat et al. [5] proposed a quantum secure data encryption protocol for Internet of Things (IoT) systems using a distributed ledger model and post-quantum key exchange techniques. The proposed algorithm was found to be suitable for the resource constrained IoT environment, thus emphasizing the significance of algorithm parametrization in embedded environment. In order to address this, a number of researchers have explored the use of post-quantum key encapsulation mechanisms for privacy preserving authentication systems, such as the quantum-resistant biometric template protection schemes of Román [6]. Although it doesn't mention blockchain, this work has shown the feasibility of the CRYSTALS-family algorithms in access control scenarios. The default signing mechanism in Hyperledger Fabric was replaced with

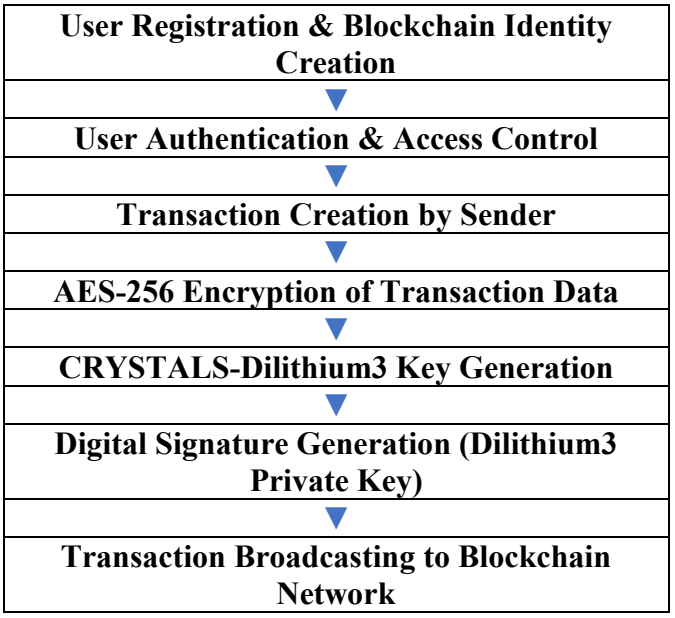
International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 10 October 2026
Page No: 5762 - 5771
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0748

CRYSTALS-Dilithium to implement the quantum-safe blockchain in Hyperledger Fabric, as suggested by Abbas et al. [7]. The experiments have shown that they are able to achieve lattice-based signature integration in permissioned blockchain networks to be feasible without any significant reduction in the throughput of the network, at least for transaction data, which, however, did not touch upon the problem of transaction data confidentiality. Reddy et al. [8] proposed a quantum-secured blockchain system for post-quantum data security, and performed extensive experimental testing on various post-quantum primitives. They showed that lattice-based algorithms are better than code-based and hash-based algorithms for the transaction signing latency of blockchain systems. Seol and Kim [9] investigated machine learning aided solutions to guarantee quantum secure blockchain availability, suggesting intelligent threat identification techniques that can detect patterns of quantum attacks and activate cryptographic algorithm switching. It is a new paradigm of adaptive security that fits with the adoption of static post-quantum algorithms. Pabla and Sultana [10] explored a modular blockchain structure, which includes quantum-safe encryption methods using key encapsulation techniques. They focused on the concept of a modular architecture as a building block for 'incremental migration' to the post quantum era, with the ability to replace components of an architecture's cryptographic algorithms without impacting the design of the whole system. This wasn't fully understood in terms of performance implications in high throughput public blockchain environments. Bose and Verma [11] gave a comprehensive overview of quantum computing and its influence on modern cybersecurity, detailing the transition in the field as vulnerabilities have been discovered in classical cryptographic algorithms and efforts are underway to standardize and implement post-quantum cryptography. The survey put blockchain security into perspective of quantum threat landscape. A relatively new study on quantum computing and blockchain security and quantum-resistant protocol design was executed in a recent conference by Vellela et al. [12]. The authors investigated the main challenges of the existing blockchain networks and suggested a structure for introducing post-quantum cryptography systematically. Sina and in [13] conducted a thorough performance analysis of quantum resistant blockchain implementations with respect to classical Secp256k1 and Schnorr signature schemes. They measured latency, throughput, and storage overhead added on by post-quantum algorithm substitution, and gave empirical benchmarks, relevant for our work. The reviewed literature as a whole show that the classical signature schemes used in blockchain are quantum vulnerable, CRYSTALS-Dilithium is the top signature candidate for the post quantum era, most of the previous works focus on replacing signatures, and comprehensive experimental evaluation of signature parametrization variants in the context of the Ethereum based blockchain is missing. This present work addresses these identified gaps with a dual layer security architecture and end-to-end performance analysis.

3. Proposed Methodology

3.1.System Architecture Overview

The suggested quantum resistant blockchain framework is a two-layer security architecture that embeds the post-quantum digital signatures CRYSTALS-Dilithium3 in combination with AES-256 symmetric encryption in an Ethereum blockchain. The overall system block diagram, showing the main processing stages and their sequential interactions, is shown in Figure 1.



International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 10 October 2026
Page No: 5762 - 5771
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0748

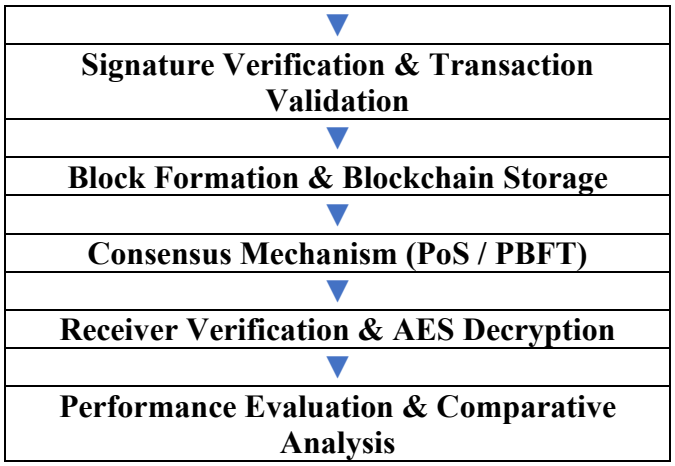


Figure 1 Proposed System Block Diagram for Quantum-Resistant Blockchain Framework

3.2.Detailed Methodology Steps

Step1: Registering the user and creating their blockchain identity [1, 5]. The system is started with an initiation of secure user registration where both sender and receiver registers secure accounts in the blockchain application. The user's registration information such as the username, password, email ID, and wallet ID is stored in the Ethereum blockchain using Solidity smart contracts. Every registered user gets a unique blockchain address that will be an unalterable identifier for all future transactions. Unlike the traditional centralized database which could be compromised, blockchain storage means that the user records are preserved in a way that makes them tamper-proof and can be verified by the transaction hash and block number.

Step 2: User authentication and access control [7, 8]. After registration, the user logs in using the login module, while the validations of the user credentials are performed by the functions of the smart contracts on the blockchain. This layer is responsible for implementing access control, which restricts access to transaction creation and verification operations to authenticated users. Impersonation attacks are reduced by session management in the web application. Blockchain authentication alongside decentralized validation reduces dependence on centralized identity providers, boosting the system's trustworthiness.

Step 3: Creation of a Transaction from the Sender [1, 13]. The transactions are initiated by authenticated senders, where the identity of the receiver, the amount of the transaction, and the metadata of the transactions are specified. These parameters are mapped to the structured transaction data object in the system. To avoid replay attacks and duplicate transactions, each transaction has a unique identifier that is cryptographically generated. This step is responsible for creating records in the blockchain format from the actions of the users, which are then moved into an encryption and signature process.

Step 4: AES-256 Encryption for Transaction Privacy [5, 10]. One of the key security features of the proposed extension is the introduction of AES-256 symmetric encryption for transaction data before putting it into the blockchain. For financial applications, in current Ethereum deployments, transaction parameters are available in plain text to any participant who knows the address of the contract - a major privacy concern. The full transaction payload is encrypted for the entire transaction with AES-256 in CBC mode and a securely generated session key. The encryption will encrypt readable transaction data into ciphertext, meaning that anyone directly accessing the contracts won't be able to read the data stored in the blockchain without the appropriate private keys.

Step 5: Key Generation based on the CRYSTALS scheme of Dilithium3 [3, 4]. This is encrypted with a system that generates an asymmetric key pair based on CRYSTALS-Dilithium post quantum cryptographic algorithm. Dilithium3 is chosen over Dilithium2 and Dilithium5 because of the empirical evaluation that it performs better in latency and key size of security category 3 (AES-192 classical security). The private key is held among the sender who uses it for signing, and the public key is spread among blockchain nodes to be used to verify. Based on the Module Learning with Errors (MLWE) lattice problem, Dilithium3 is proven to be resistant to both classical and quantum attackers.

Step 6: Generate Digital Signature [4, 7]. The AES-encrypted transaction data is digitally signed using sender's Dilithium3 private key through the deterministic signing algorithm. This signature provides cryptographic proof of origin, and ensures the authenticity of transactions (one-bit change in the encrypted payload results in a failure of signature

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 10 October 2026
Page No: 5762 - 5771
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0748

verification). The signed transaction packet contains the encrypted transaction data, sender's public key, transaction's time, and the digital signature from the Dilithium3 protocol, which creates a complete, self-contained authenticated message bundle.

Step 7: Transaction Broadcasting to Blockchain Network [1, 8]. The signed transaction packet is then sent to all the nodes that are participating in the blockchain through the smart contract call on Ethereum. The broadcasting module is responsible for distributing the encrypted transaction data and related Dilithium3 signatures to the peer-to-peer network. Only ciphertext is broadcasted which brings transparency and synchronization throughout the network and retains data confidentiality. This architecture is server independent and provides a completely distributed transaction processing environment.

Step 8: The signature verification and transaction validation step [4, 9, 13] is performed. Once the transaction is received, blockchain nodes will verify the signature with the sender's public key through Dilithium3. The verification algorithm recalculates the verification polynomial and checks it to the received signature components. If the verification is successful, the transaction is considered to be valid, otherwise it is rejected. The validation process gives the quantum adversary long-term quantum-safe assurance of transaction integrity and non-repudiation, as the verification of Dilithium3 is still not feasible to compute.

Step 9: Block Formation and Blockchain Storage [5, 12]. The validated transactions are aggregated into blocks and permanently stored on the blockchain. Every block contains data for the transactions that are encrypted with AES, signed with Dilithium3, hashed with SHA-256, timestamped, and contains a hash of the previous block. Global tamper proof: if any block is altered, it will invalidate all subsequent block hashes, because this relies on the hash-linked chain structure. In addition, AES encryption and the immutability of hash-chaining guarantees both confidentiality and integrity at the same time, which is more than the conventional plaintext blockchain storage does.

Step 10: Consensus Mechanism and Block Finalization [8, 10]. After the blocks are formed, the network then uses a consensus mechanism to reach a distributed agreement on the validity of the newly formed block, the different deployment configurations involve using either a Proof-of-Stake (PoS) consensus mechanism or a Practical Byzantine Fault Tolerance (PBFT) consensus mechanism. After reaching an agreement, the block is permanently added to the blockchain ledger and shared by all nodes involved. This process ensures that double-spending is prevented, block insertion is not allowed, and fraudulent transaction finalization is not possible.

Step 11: Verify and decrypt receiver transactions [3, 7]. The verification module is accessed by authorized receivers to validate transactions received by performing two successive operations. The sender's public key is used to verify the signature Dilithium3, ensuring authenticity and integrity. Second, when the verification is successful, AES-256 decryption is employed to decrypt transaction data from the pre-shared session key, and obtain the original plaintext transaction data. The sequential verify-then-decrypt architecture allows only for decryption of verified and unaltered transactions, thwarting disclosure of unauthorized data and cryptographic oracle attacks.

Step 12: Performance Evaluation and Comparative Analysis [1, 13]. The proposed system is analyzed quantitatively for performance parameters like latency due to the time taken for a transaction (in milliseconds) and key size (in bytes). The proposed approach is then compared experimentally to the RSA-2048, ECC (Secp256k1), Dilithium2 and Dilithium3 algorithms, to attempt to characterize the tradeoff between security and performance. Latency trend graphs and key size bar charts are used to visualize the results to assist in comparison.

3.3.Ethereum Smart Contract Design

The proposed solution is a smart contract written in Solidity that is deployed on the Ethereum testnet to handle user identities on a decentralized manner and storage of encrypted transactions. Two main data structures are outlined in the contract: a User struct contains the username, a hash of the password, the email address, and the wallet address; and a Transaction struct contains an encrypted payload, Dilithium3 signature bytes, the sender's public key, the time, and a flag indicating whether the transaction is verified. The contract has the following main

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 10 October 2026
Page No: 5762 - 5771
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0748

functions: registerUser() creates a user account on the blockchain and maps its address to a user; authenticateUser() validates a credential and returns the user's account status; createTransaction(), submits an encrypted transaction to the blockchain and assigns a key to store the transaction's signature; verifyTransaction(), retrieves the signature from the blockchain and updates the user's status; and getTransaction(), retrieves a valid encrypted transaction by a credential. Smart contract deployment is done via truffle migration scripts from the development environment of Ganache Ethereum and the address of the deployed contract will be stored in the application which will be later used for RPC calls.

4. Results and Analysis

4.1.Experimental Setup

The suggested quantum-resistant blockchain system has been implemented and tested in a simulated environment of the Ethereum blockchain development process. The software stack included Python 3.7 as the main programming language, the dilithium-py library to operate on the CRYSTALS-Dilithium3, pycryptodome library for AES-256 encryption, Web3.py to communicate with the Ethereum blockchain, and Flask as the lightweight web application server. Using Ganache with ten default accounts that were pre-funded, the Ethereum blockchain was instantiated for testing this. The smart contracts were compiled and deployed by Truffle framework v5. x. To describe the latency and throughput scaling behavior, experimental transactions were programmed from 10 to 100 transactions within the range.

4.2.Module Descriptions

The source code of the smart contract (Solidity) which includes user registration and functions for managing transactions, using AES encrypted data storage. After running the Truffle migration command, the accounts and private keys for Ganache Ethereum are shown and the contract address is printed in the console window Figure 2 shows that the contract has been successfully deployed, and the address of the deployed contract is also stored. Shown in Figure 3

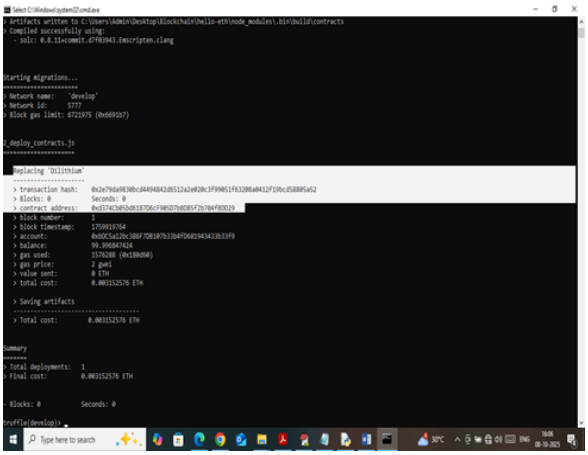


Figure 2 Smart contract Deployment

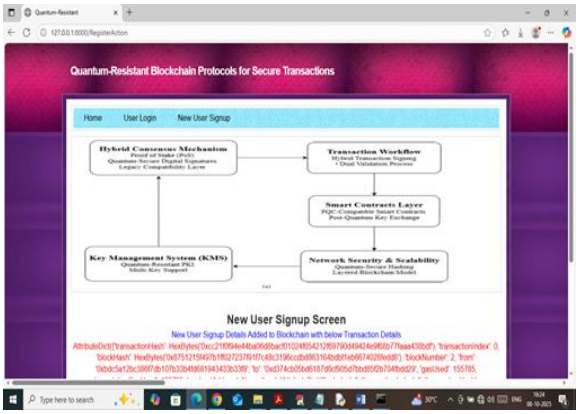


Figure 3 Post-registration confirmation screen

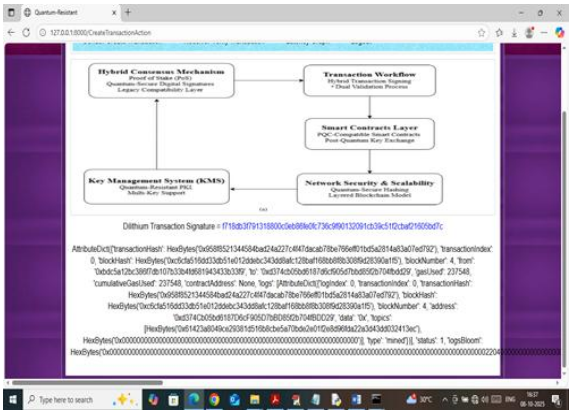


Figure 4 Dilithium3 digital signature

There are four main modules, with navigation links to each on the application landing page, after user sign-up form which has input boxes for a username, password, email, and wallet identity. The post-registration confirmation screen shows details of the Ethereum blockchain transactions (in red text) that were conducted, along with the block number,

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 10 October 2026
Page No: 5762 - 5771
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0748

transaction hash, gas usage and contract address, indicating successful user record storage on-chain. After giving details in user login screen. The authenticated user dashboard with transaction management navigation options along sender enters the identity of the receiver and the amount of the transaction in the transaction creation. The transaction confirmation screen (Figure 4(c is presented with the Dilithium3 digital signature in blue, and the Ethereum blockchain transaction metadata in black, indicating that the transaction was successfully stored as encrypted data. Shown in Figure 5

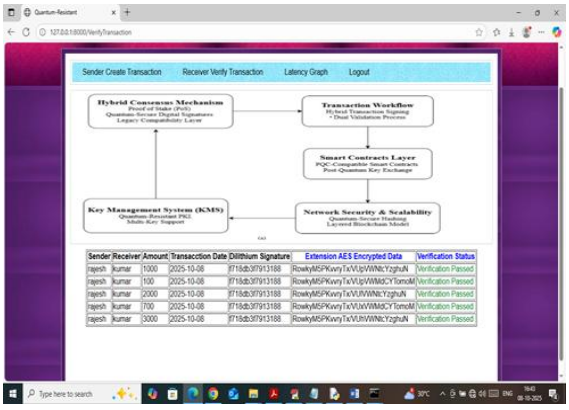


Figure 5 Signature column that contains the Dilithium3 signature

The receiver login screen is displays while the receiver dashboard screen shows the receiver transaction verification table, which contains four transaction attribute plaintext columns, the AES-encrypted data column, and the signature column that contains the Dilithium3 signature. Note that unauthorized observers who access the blockchain records see only ciphertext in the AES-encrypted data column. The last verification status column confirms that all transactions retrieved are valid.

4.3.Latency Performance Analysis

The latency comparison graph shown in Figure 6 for various number of transactions 10, 20, 40, 60, 80 and 100 between the two algorithms: RSA and the proposed CRYSTALS-Dilithium3. The x axis is number of transactions processed and the y axis is the accumulated latency in milliseconds. The blue line shows latency of RSA-2048 and the orange line shows the latency of CRYSTALS-Dilithium3.

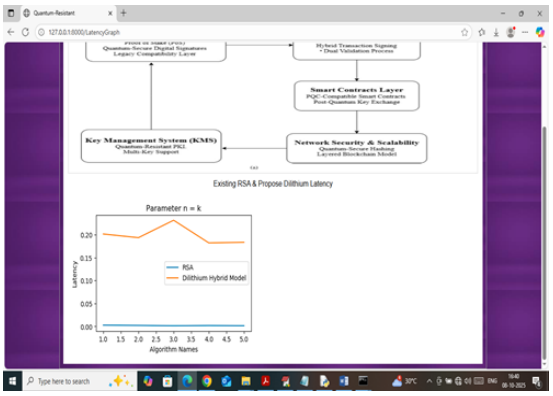


Figure 6 Latency Comparison

As a whole, the experimental results show that the use of RSA-2048 produces lower absolute latency in all transaction counts, due to simplicity of the operations in the keys and decades of optimisation of hardware and software. CRYSTALS-Dilithium3 has a higher transaction latency as it uses lattice polynomial arithmetic for signed and verification. With higher transaction counts, however, the gap between the latencies becomes smaller, as the overhead of the Python runtime takes effect, and the latency of Dilithium3 is almost linear, indicating predictable latency that is suitable for production use of the blockchain. But the key takeaway is that extra latency that Dilithium3 adds to RSA is reasonable trade-off between security and performance: RSA-2048 is still susceptible to Shor's algorithm on fault-tolerant quantum computer, which means that the lower latency is no help to long-lived blockchain ledgers.

4.4.Dilithium2 vs. Dilithium3

Dual-graph comparison of Dilithium2 and Dilithium3 is shown in Figure 7. The first graph (left panel) shows latency scaling: the x-axis represents number of transactions and the y-axis represents latency in milliseconds. The green line represents Dilithium2 and blue line represents Dilithium3. Interestingly, the latency of Dilithium3 is lower than Dilithium2, which can be explained as using a more efficient polynomial arithmetic in the higher-parametrization version with optimised NTT (Number Theoretic Transform) operations. The second graph (on the right) displays an important size comparison, again as a bar chart, and with the names of the algorithms shown on the x-axis and the sizes of the keys (in bytes) on the y-axis.

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 10 October 2026
Page No: 5762 - 5771
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0748

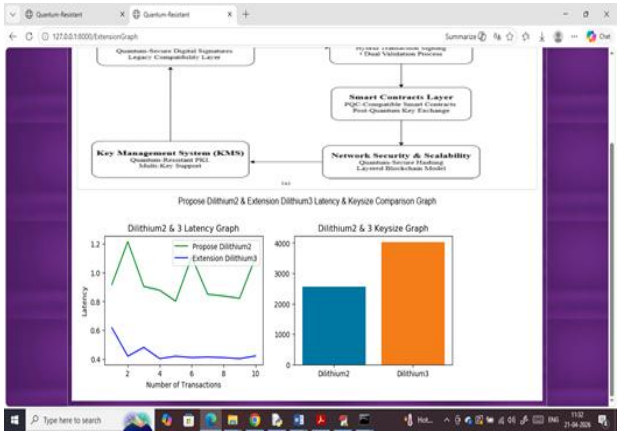


Figure 7 Latency and Key size graph

The public keys of Dilithium3 are significantly larger than those of Dilithium2 as well as its private keys: Dilithium2 public key size is 1312 bytes, Dilithium3 public key is 1952 bytes and Dilithium2 private key is 2528 bytes, while Dilithium3 private key is 4000 bytes. The increased key size of Dilithium3 is directly proportional to the increase in its security category (AES-192 compared to AES-128 for Dilithium2), which means that classical and quantum attackers cannot computationally break the key in both cases. The low latency and high security of Dilithium3 over Dilithium2 and the manageable key sizes over Dilithium5 further justify the choice of Dilithium3 for the proposed blockchain application as the optimal parameter size.

4.5.Comparative Performance Table

The results of performance and security comparison are summarized in table 1 for the algorithms studied, namely RSA-2048, ECC (Secp256k1), Dilithium2 and Dilithium3. As shown in Table 1, both RSA and ECC have quantum-vulnerable public key sizes and latencies. Dilithium2 and Dilithium3 are both based on MLWE (lattice hardness), and have quantum resistance. Dilithium3 is preferred parametrization as it offers lower latency than Dilithium2, while supporting larger key sizes. The size of the key is larger in Dilithium3 than in classical algorithms - but this is deemed a reasonable storage cost when the security guarantees of blockchain applications are a critical concern.

Table 1 The Comparative presentation of performance and security of cryptographic algorithms

Algorithm	Security Basis	Latency	Public Key Size	Key Overhead	Quantum Safety
RSA-2048	Integer Factorization	Low	~256 bytes	High	Quantum Vulnerable
ECC (Secp256k1)	Discrete Log (EC)	Low	~64 bytes	High	Quantum Vulnerable
Dilithium2	MLWE (Lattice)	Medium	1,312 bytes	Medium	Quantum Resistant
Dilithium3	MLWE (Lattice)	Low-Medium	1,952 bytes	Low-Medium	Quantum Resistant ✓

4.6.Comparison with Related Works

Table 2 is the qualitative comparison of the proposed framework with the representative works related to the quantum-resistant blockchain from the research field.

Table 2 Comparison of Proposed Framework with Related Works

Ref.	Platform	PQ Algorithm	Data Encryption	Signature Replace	Param. Comparison	Deployment
Abbas et al. [7]	Hyperledger Fabric	Dilithium	No	Yes	No	Permissioned
Prajapat et al. [5]	IoT Blockchain	PQ-KEM	Yes	No	No	IoT Nodes

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 10 October 2026
Page No: 5762 - 5771
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0748

Reddy et al. [8]	Generic BC	Dilithium	No	Yes	No	Testnet
Sinai & In [13]	Generic BC	PQ Schemes	No	Yes	No	Testnet
Proposed Work	Ethereum	Dilithium3+AES	Yes	Yes	Yes	Ethereum

Conclusion and Future Scope

This paper introduced novel quantum-resistant blockchain framework to solve dual security problems from the vulnerabilities of quantum-resistant signatures and the confidentiality of the transaction data in decentralized systems based on Ethereum. The proposed architecture combines the lattice-based digital signatures of CRYSTALS-Dilithium3 with the symmetric encryption of AES-256 in a complete twelve-step transaction pipeline ranging from user registration to user authentication, transaction creation, transaction encryption, generation of public and private keys, digital signing, broadcasting, transaction verification, block building, consensus, decryption, and performance evaluation. The experimental evaluation showed that CRYSTALS-Dilithium3 offers provably quantum-resistant transaction authentication with linearly-scalable latency characteristics. After comparing the two, it was shown that Dilithium3 achieves better security properties and lower latency for the same key size than Dilithium2, and therefore is the optimal parametrization for blockchain deployment. It provides privacy to transactions that were missing from previous post-quantum blockchain research, as well as the ability to prevent unauthorized blockchain observers from seeing the transactions in plaintext, thanks to the AES-256 encryption layer. The proposed framework meets all the stated goals: decentralized user identity management, end to end encrypted transaction privacy, quantum-resistant signature authentication and smart contract-based blockchain storage with tamper evident hash-linking. The system is validated by implementing it in functional Python and Ethereum, and is quantitatively evaluated with regards to the three algorithms mentioned above: RSA-2048, ECC, Dilithium2, and Dilithium3. The Further integration of CRYSTALS-Kyber (FIPS 203) for quantum-safe key encapsulation, which will replace the pre-shared AES session key distribution with a completely post-quantum key agreement protocol, and remove the last classical vulnerability from the current architecture. Testing the proposed framework on public mainnet of Ethereum or L2s to understand the gas cost impacts of storing Dilithium3 signatures and AES ciphertext in smart contract storage, which will help guide the feasibility of practical deployment. Extension of hash-based signature schemes to be supported by SPHINCS+ as a stateless alternative to Dilithium, with different security assumptions (hash functions resistant to collisions instead of MLWE) for contexts that need to rely on multiple cryptographic bases. Explore zero-knowledge proof (ZKP) solutions, such as zk-SNARKs or zk-STARKs, to facilitate the verification of transactions while maintaining confidentiality, thus enhancing the assurances of confidentiality. Development of the hardware accelerated implementation using FPGA or dedicated post-quantum cryptography co-processors to minimize the latency overhead of the Dilithium3 lattice arithmetic, which is going to be useful for applications involving signing operations on financial blockchain platforms requiring high throughput. A security analysis covering the long-term security of AES-256 against quantum computing using Grover's algorithm to break the primitive, which might encourage the adoption of AES-256 with doubled key size or the use of quantum safe symmetric primitives as they are standardized.

References

[1]. Puneyani, V., & Bhat, K. V. (2025). Quantum-Resistant Blockchain Protocols for Secure Transactions. IEEE Access, 13, 108984–108991.
[2]. Fedorov, A. K., Kiktenko, E. O., & Lvovsky, A. I. (2018). Quantum computers put blockchain security at risk. Nature, 563(7732), 465–467.
[3]. Bernstein, D. J. (2025). Post-quantum

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 10 October 2026
Page No: 5762 - 5771
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0748

cryptography. In Encyclopedia of Cryptography, Security and Privacy (pp. 1846–1847). Cham: Springer Nature Switzerland.

[4]. Alagic, G., Apon, D., Cooper, D., Dang, Q., Dang, T., et al. (2022). Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process. NIST IR 8413.

[5]. Prajapat, S., Kumar, N., Das, A. K., Kumar, P., & Ali, R. (2025). Quantum-safe blockchain-assisted data encryption protocol for internet of things networks. Cluster Computing, 28(1), 5.

[6]. Román, R. A. (2022). A Quantum-Resistant Face Template Protection Scheme using Kyber and Saber Public Key Encryption Algorithms. In BIOSIG 2022 (pp. 23–33). Gesellschaft für Informatik eV.

[7]. Abbas, S., Sultana, A., & Kaddoum, G. (2024). Quantum-safe blockchain in hyperledger fabric. IEEE Networking Letters, 7(1), 61–65.

[8]. Reddy, N. R., Suryadevara, S., Reddy, K. G. R., Umamaheswari, R., Guttula, R., & Kotoju, R. (2025). Quantum secured blockchain framework for enhancing post quantum data security. Scientific Reports, 15(1), 31048.

[9]. Seol, J., & Kim, J. (2025). Machine learning ensures quantum-safe blockchain availability. Journal of Computer Information Systems, 65(5), 555–579.

[10]. Pabla, T., & Sultana, A. (2024). Modular blockchain architecture: Securing data with quantum-safe encryption. In 2024 International Conference on Quantum Communications, Networking, and Computing (QCNC) (pp. 198–203). IEEE.

[11]. Bose, B., & Verma, S. (2026). Impact of Quantum Computing on Modern Cybersecurity: Challenges, Opportunities, and the Evolution Toward Post-Quantum Cryptography. From Bits to Qubits: The Quantum Transformation of Computing, 173–210.

[12]. Vellela, S. S., Vuyyuru, L. R., Jidugu, S. K., Rao, M. P., & Srinivas, B. R. (2025). The Impact of Quantum Computing On Blockchain Security and Quantum Resistant Protocols. In 2025 2nd International Conference on Intelligent Systems for Cybersecurity (ISCS) (pp. 1–6). IEEE.

[13]. Sinai, N. K., & In, H. P. (2024). Performance evaluation of a quantum-resistant Blockchain: a comparative study with Secp256k1 and Schnorr. Quantum Information Processing, 23(3), 99.