IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 22 July 2025, accepted 8 August 2025, date of publication 18 August 2025, date of current version 27 August 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3600068

METHODS

Hybrid Quantum-Safe Cryptographic Scheme With Secure Key Exchange and Signature Scheme

PERERA K. MADUNI[1], ILMU BYUN[2], (Member, IEEE), JEONGIL SEO[1], (Member, IEEE), AND KYEONGJUN KO[3], (Member, IEEE)

[1]Department of Computer Engineering, Dong-A University, Busan 49315, South Korea
[2]Korea Railroad Research Institute, Uiwang-si, Gyeonggi-do 16105, South Korea
[3]School of Electronics and Electrical Engineering, Hongik University, Seoul 04066, South Korea

Corresponding authors: Ilmu Byun (ilmubyun@krri.re.kr) and Jeongil Seo (jeongilseo@dau.ac.kr)

This work was supported in part by the Research and Development Program (Private 5G-Railway Core Technology Development for Railway Digital Transformation) of Korea Railroad Research Institute under Grant PK2504D1, and in part by the Dong-A University Research Fund.

ABSTRACT Hybrid cryptographic protocol was proposed by integrating quantum-safe algorithms (Kyber and Dilithium) together with a classical ECDSA-based signature scheme to verify secure proof on blockchain. Since quantum computers pose some loopholes in existing cryptography algorithms, the proposed mechanism addresses those vulnerabilities by maintaining compatibility with the existing infrastructure. The protocol integrates post-quantum key exchange to derive shared secrets securely. The shared secrets are used to enable hybrid encryption for efficient data protection. Additionally, the implementation uses a dual-signature approach, using quantum safe and classical signatures. These signatures are combined to ensure robust and tamper-resistant proof verification. The deployed solution validates its feasibility and efficiency for smart contract-based verification. The protocol maintains backward compatibility with existing systems while providing strong resistance to quantum attacks. Benchmark results demonstrate the system's scalability and robustness under high transaction loads, with efficient gas usage for on-chain proof validation. This approach paves the way for post-quantum secure applications, including verifiable proofs, secure key exchanges, and decentralized systems resilient to quantum adversaries.

INDEX TERMS Quantum-resistant cryptography, hybrid encryption, blockchain, smart contract, Merkle root.

I. INTRODUCTION

For a long time, traditional cryptographic systems have been the backbone of secure communication, authentication, and data integrity. However, traditional cryptography has faced significant challenges with the rapid development of quantum computers. Algorithms such as Rivest–Shamir–Adleman (RSA), Elliptic Curve Digital Signature Algorithm (ECDSA), and Elliptic Curve Cryptography (ECC) rely on the mathematical hardness of problems like integer factorization or discrete logarithms [1]. However, quantum algorithms such as Shor's algorithm can solve these problems in polynomial time [2]. With further improvement of quantum computers in the future, these classical algorithms will no longer provide adequate security, putting a wide range of applications, including secure transactions and identity management, at risk.

In response to this threat, the field of quantum-safe cryptography has developed algorithms that resist quantum attacks. These algorithms are based on mathematical problems that are difficult for both classical and quantum computers. Lattice-based cryptographic schemes such as Kyber and Dilithium are among the most prominent post-quantum algorithms recommended by NIST (see Section IV for details) [3], [5].

Blockchain technologies have revolutionized industries such as finance, healthcare, and supply chain by offering decentralized, transparent, and immutable data management.

The associate editor coordinating the review of this manuscript and approving it for publication was Peng-Yong Kong.

Blockchain security, by design, relies on classical cryptographic techniques, such as ECDSA for digital signatures and Diffie-Hellman for key exchange, which is used on popular platforms such as Ethereum and Bitcoin. Blockchain systems provide secure transactions, validate identities, and ensure data integrity. ECDSA and Diffie-Hellman algorithms are secure against classical computing attacks and are vulnerable to quantum algorithms. Therefore, they cannot ensure their own integrity in the near future. As such, there is a pressing need to integrate quantum-safe cryptography into blockchain infrastructures to ensure long-term security and compatibility with the evolving threat landscape.

Despite advances in quantum-safe cryptography, integrating these algorithms into blockchain systems remains a challenge. Most quantum-safe techniques are not directly compatible with blockchain implementations due to differences in performance characteristics, key sizes, and signature structures. Furthermore, existing blockchain infrastructures have not been designed to incorporate hybrid cryptographic systems that leverage both classical and post-quantum security measures.

Also, the lack of real-world implementations that demonstrate the combination of quantum-safe cryptography with smart contracts for proof verification poses a major gap. Smart contracts must be able to handle post-quantum signatures and key encapsulations efficiently without compromising the decentralized and transparent nature of the blockchain [2], [3], [5], [10].

Therefore, the main objective is to design and implement a hybrid cryptographic framework addressing the vulnerabilities of blockchain systems in a post-quantum era by combining classical and quantum-safe algorithms [3], [4], [5], [10].

Merkle roots are critical in blockchain systems for efficient verification of large datasets. By aggregating multiple signatures into a single root, they enable compact and tamper-resistant validation. In the proposed system, Merkle roots combine classical and quantum-safe signatures (ECDSA and Dilithium), allowing efficient on-chain proof verification without compromising security [11].

Modern cybersecurity systems should adopt post-quantum cryptographic (PQC) mechanisms to mitigate the emerging risks posed by quantum computing. Our blockchain-integrated hybrid framework combines classical and quantum-safe algorithms, leveraging Kyber for key encapsulation and Dilithium for digital signatures. Unlike conventional models reliant on centralized Certificate Authorities (CAs), our approach ensures decentralized, tamper-proof key verification, improving long-term trust and transparency.

To address the limitations of current quantum-safe protocols like CECPQ2 and TLS 1.3 with PQC extensions, we adopt a fully post-quantum architecture. This eliminates reliance on elliptic curve primitives and enhances resistance to quantum decryption threats. A detailed comparative analysis against these solutions is provided in Section VII.

Our encryption model employs AES for fast symmetric encryption and Kyber1024 to protect the AES session key, offering post-quantum confidentiality for large-scale data. Signature generation is performed using Dilithium5, providing strong security guarantees while maintaining computational efficiency. Section IV and Section VII detail the algorithm selection and benchmarking results.

The primary contribution of this paper is the design and implementation of a novel hybrid cryptographic framework that seamlessly integrates Kyber1024 and Dilithium5 with Ethereum-based smart contracts. We propose a dual-signature Merkle-root scheme to maintain compatibility with legacy blockchain infrastructure while embedding strong post-quantum guarantees. Extensive performance and security evaluations demonstrate the superiority of our system over existing transitional solutions such as TLS 1.3 + PQC and CECPQ2.

The remainder of this paper is organized as follows. Section II and Section III introduce the recent blockchain-based post-quantum results and the system model, respectively. The key generation and exchange, encryption and decryption, signing and verification algorithms are discussed in Section IV. Section V presents how smart contracts are interacted to this project, and the potential applications of the algorithm are mentioned in Section VI. Then, the simulation results are presented in Section VII. Finally, conclusion and future work are discussed in Section VIII.

II. RELATED WORKS

Several research efforts have explored the integration of post-quantum cryptography (PQC) into blockchain systems. This section outlines the most relevant contributions and positions our work within the current state of the art.

Castiglione et al. proposed a framework that integrates Falcon signatures into blockchain systems to secure low-cost IoT devices [6]. Their approach emphasizes lightweight digital signatures and performance tuning for constrained environments. However, it does not address hybrid encryption or decentralized key validation mechanisms, which are key components of our framework. Furthermore, our solution uses Dilithium5 and Kyber1024, which have recently been standardized by NIST as ML-DSA and ML-KEM respectively.

Holmes studied the impact of PQ signatures on blockchain and distributed ledger technologies (DLT) [7]. The work highlights practical deployment issues, including signature size and computational overhead. In contrast, our implementation offers a Merkle root–based signature verification mechanism to reduce on-chain data size and mitigate gas cost implications, thus addressing these challenges more effectively.

Fernández-Caramés and Fraga-Lamas provided a comprehensive review of blockchain cryptographic mechanisms resistant to quantum attacks [5]. Their work is primarily conceptual, offering design principles and threat models. Our study builds on their insights by presenting a complete

end-to-end implementation and performance analysis, with empirical benchmarks comparing our approach against TLS 1.3 + PQC and CECPQ2.

Ismail et al. provided a comprehensive review of blockchain cryptographic mechanisms resistant to quantum attacks [8]. Their work is primarily conceptual, offering design principles and threat models. Our study builds on their insights by presenting a complete end-to-end implementation and performance analysis, with empirical benchmarks comparing our approach against TLS 1.3 + PQC and CECPQ2.

Zhang et al. proposed a certificateless ring signature scheme tailored for IoT networks, aimed at enhancing user privacy and reducing reliance on centralized certification authorities [9]. Their focus on privacy preservation through lightweight cryptographic constructs addresses key IoT security concerns. However, their model does not consider hybrid post-quantum encryption or Ethereum-compatible smart contract interaction for proof verification. Our proposed framework complements such privacy-preserving schemes by offering tamper-proof authentication and verifiability via decentralized blockchain proofs.

III. SYSTEM MODEL

The entire mechanism of the proposed framework is shown in Figure 1. The proposed system integrates a hybrid cryptographic framework utilizing both quantum-safe and classical cryptographic techniques to ensure secure data transmission, verification, and storage in blockchain applications. It employs Kyber for secure key exchange, Dilithium and ECDSA for digital signatures, and AES encryption for data confidentiality. The Ethereum blockchain is used to store and verify cryptographic proofs, enhancing security and transparency [16].

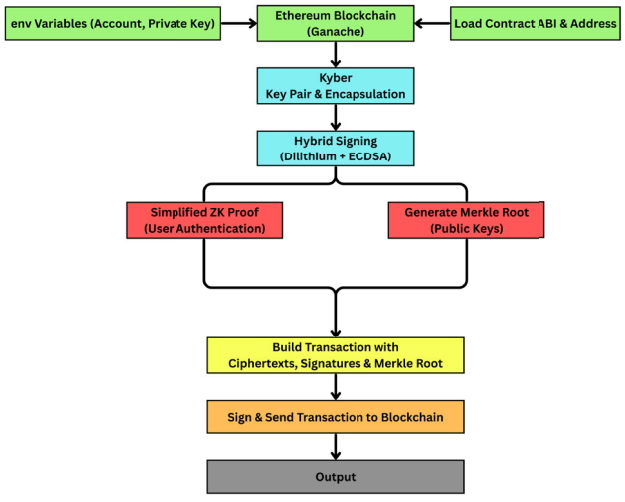


FIGURE 1. Proposed mechanism.

As per Figure 2, during Key Generation and Encapsulation, the sender generates a Kyber1024 keypair and signs the

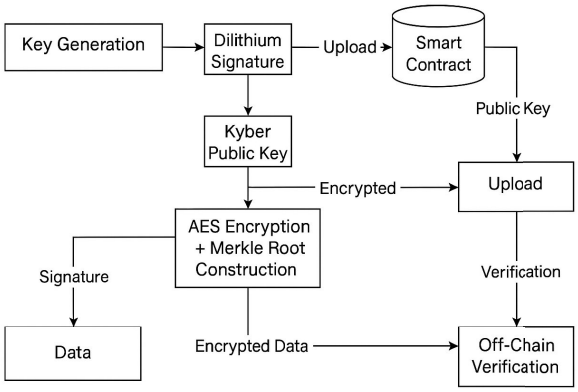


FIGURE 2. Overview of the proposed workflow: from key generation and encryption to blockchain-based verification.

generated public key using their Dilithium5 private key. This signed public key, along with the raw Kyber key, is transmitted to the receiver. The receiver then verifies the signature using the sender's Dilithium5 public key, which is retrieved from the Ethereum smart contract where it was permanently registered during the initial key publication phase. This decentralized and tamper-resistant verification ensures the authenticity of the Kyber public key, preventing man-in-the-middle (MITM) attacks by rejecting forged public keys. Upon successful verification, the receiver encapsulates a shared secret, derives a hybrid shared key using SHA-256, and proceeds with hybrid encryption.

During hybrid encryption the shared secret obtained from key encapsulation is used to generate an AES key. Data is encrypted using AES-CBC mode for confidentiality. A random initialization vector (IV) is generated and included with the cipher-text. PKCS#7 padding ensures that plain-text fits the block size of AES. The encrypted data, IV, and authentication tag are bundled together before transmission.

Throughout the hybrid signing process, the sender signs the AES-encrypted data using Dilithium, a post-quantum digital signature algorithm. The signature ensures authenticity and non-repudiation. A redundant ECDSA signature is generated for compatibility with blockchain verification tools. This ensures interoperability with existing Ethereum smart contracts. The signatures from Dilithium5 and ECDSA are structured into a Merkle tree. The Merkle root, derived using SHA-256, is stored on the blockchain. Any party can verify data integrity by recomputing the Merkle root using the stored signatures [16].

A smart contract on Ethereum is integrated to store public keys and proofs. The contract verifies that the Merkle root, signatures, and encrypted data match the expected values. The Dilithium and ECDSA signatures are checked to ensure authenticity. Ganache is used as a local Ethereum test network. The client encrypts the data using AES. The encrypted data is signed using Dilithium and ECDSA. The Merkle root is computed and stored on the blockchain.

A smart contract verifies the signatures and integrity of the data.

The hybrid cryptographic framework ensures quantum-safe key exchange, hybrid encryption, and signature verification. By leveraging blockchain for proof storage and validation, the system provides enhanced security, integrity, and verifiability in decentralized applications [16].

A. POST-QUANTUM ALGORITHM STANDARDIZATION

In recent developments, the cryptographic algorithms utilized in our proposed framework, Kyber-1024 and Dilithium-5 have been formally standardized by the U.S. National Institute of Standards and Technology (NIST) as ML-KEM (Module Lattice–Key Encapsulation Mechanism) and ML-DSA (Module Lattice–Digital Signature Algorithm), respectively. These standards are published as FIPS 203 [14] and FIPS 204 [15], and they represent the culmination of several years of research into post-quantum secure algorithms.

The standardized versions introduce vital updates, including recommended implementation techniques to resist side-channel attacks (SCAs) and fault attacks (FAs), which were potential vulnerabilities in earlier drafts. Despite of the updates, our protocol structure is fully compatible with ML-KEM and ML-DSA because the underlying mathematical structure and operational semantics remain consistent between these versions, requiring only minor adjustments for full compliance.

Moreover, we acknowledge the significance of widely trusted cryptographic libraries such as PQClean and liboqs. Both libraries now offer high-assurance implementations of ML-KEM and ML-DSA. These libraries not only align with the FIPS standards but also provide modular structures for integrating these primitives into broader security architectures such as the hybrid cryptographic system proposed in this study.

B. THREAT MODEL

This section formally outlines the threat assumptions and adversarial capabilities considered in the design of the proposed cryptographic system.

1) ADVERSARIAL CAPABILITIES

The adversary is assumed to have the following powers:

- Man-in-the-Middle (MITM): Full control over the communication channel, with the ability to intercept, modify, or inject data packets.
- Replay Attacks: Capability to capture and resend previously valid messages in an attempt to deceive the system.
- Key Extraction Attempts: Access to public blockchain data, including encrypted symmetric keys, public keys, and Merkle roots.
- Signature Forgery Attempts: Efforts to generate fraudulent messages or signatures using classical or quantum resources.

Algorithm 1 Key Generation and Exchange Module With Authentication

1: Generate asymmetric keypair using post-quantum KEM (e.g., Kyber)
2: Generate digital signature keypair using post-quantum signature scheme (e.g., Dilithium)
3: Sign the public key of KEM using the private signing key
4: Transmit both KEM public key and its signature to the recipient
5: Recipient verifies the signature using sender's public signing key
6: Retrieve sender's Dilithium5 public key from Ethereum smart contract
7: **if** `valid` **then**
8: encapsulate a shared secret using the received public key
9: Store `ciphertext` and `shared_secret` for encryption/decryption
10: **else**
11: terminate protocol due to authentication failure
12: **end if**

2) MITIGATION STRATEGIES

The proposed system addresses the above threats through:

- Post-Quantum Signatures: The use of Dilithium5 ensures resilience against both classical and quantum attacks on digital signatures.
- Hybrid Dual Signature Verification: The inclusion of ECDSA alongside Dilithium enables compatibility with current Ethereum platforms while reinforcing trust via a Merkle root anchored on-chain.
- Transaction Freshness: Ethereum's built-in nonce mechanism thwarts replay attacks by enforcing unique transaction IDs.
- Secure Key Encapsulation: Kyber1024 guarantees secure symmetric key exchange even under quantum adversarial assumptions.

IV. KEY GENERATION AND EXCHANGE, ENCRYPTION AND DECRYPTION, SIGNING AND VERIFICATION MODULES

Key generation and exchange algorithm leverages the Kyber1024 key encapsulation mechanism, which is a part of the NIST post-quantum cryptography standardization process. This algorithm is critical in establishing secure communication channels, particularly in environments where quantum computing poses a significant threat to traditional cryptographic schemes like RSA and ECC.

A. MODULE 1: KEY GENERATION AND EXCHANGE

Module 1 describes the proposed key generation and exchange process using the Kyber1024 key encapsulation mechanism.

To address the inherent lack of authentication in Kyber-based KEMs and mitigate the risk of man-in-the-middle (MITM) attacks, the proposed system introduces a public key authentication phase before key encapsulation. Each party signs own Kyber1024 public key using own certified Dilithium5 private key. The signed Kyber public key is then transmitted to the peer along with the unsigned Kyber key. Upon receipt, the recipient verifies the signature using the sender's Dilithium5 public key retrieved from the Ethereum smart contract where it was stored during the initial key registration phase. This ensures decentralized, verifiable, and tamper-resistant authentication of key material. If verification fails, the key exchange is aborted. If verification succeeds, the recipient proceeds with encapsulating the shared secret using the verified Kyber key.

After initialization, the system generates a pair of public and private cryptographic keys, which are essential to securely manage the encapsulation and decapsulation processes. The public key is used to encapsulate shared secrets, while the private key enables decapsulation, thereby preserving confidentiality during data exchange.

To augment security further, following successful verification of public key authenticity, the protocol integrates an AES symmetric key. AES is employed to ensure the confidentiality of data throughout the transmission phase. This hybrid encryption strategy combines the advantages of asymmetric encryption provided by Kyber and symmetric encryption capabilities offered by AES, resulting in secure yet efficient data communication. During encapsulation, the public key facilitates the generation of ciphertext along with a shared secret. This ciphertext can be safely transmitted over unsecured communication channels without exposing the AES key, maintaining its confidentiality unless the private key is compromised. Subsequently, both ciphertext and the shared secret are securely stored, ready to support future encryption and decryption operations.

Utilizing Kyber1024 in key-exchange protocols marks significant progress in the development of post-quantum cryptography. Its primary contribution is enhanced security against quantum computing attacks, addressing weaknesses inherent in traditional cryptographic algorithms susceptible to quantum-based cryptanalysis. The quantum resilience of Kyber1024 is crucial for ensuring long-term security within communication infrastructures as quantum computing continues to advance. Additionally, the hybrid encryption technique provided by Kyber1024 balances robust security with optimal performance, making it particularly effective in contexts requiring rapid data processing without sacrificing security levels.

Kyber1024's versatility extends notably to blockchain applications, where secure key-exchange mechanisms,when combined with authenticated public key verification via Dilithium5, underpin transactional integrity and authenticity. This layered approach significantly increases the reliability and trustworthiness of blockchain-based platforms.

Algorithm 2 Encryption and Decryption Module

1: Derive symmetric encryption key from shared secret
2: Encrypt the plaintext using a symmetric encryption scheme (e.g., AES)
3: Transmit encrypted data and encapsulated key
4: Receiver decapsulates the shared secret using private KEM key
5: Decrypt the ciphertext using derived symmetric key

Furthermore, despite offering rigorous security, Kyber1024 maintains computational efficiency, making it highly applicable in scenarios involving extensive deployments such as Internet of Things (IoT) networks, cloud infrastructure, and decentralized systems. Given that Kyber has been adopted in FIPS 203 by NIST as part of post-quantum cryptographic standards, integrating it into security frameworks not only enhances interoperability but also promotes standardization across various cybersecurity implementations [18].

B. MODULE 2: ENCRYPTION AND DECRYPTION

Module 2 describes the encryption and decryption process using the shared secret from the key exchange.

The Encryption and Decryption Module serves as a cornerstone in securing data transmission, especially in environments integrating post-quantum cryptographic mechanisms. This Module ensures data confidentiality through hybrid encryption techniques that synergize the efficiency of symmetric encryption (AES) with the robustness of asymmetric cryptography (Kyber). The encryption and decryption process comprises five critical steps, each playing a pivotal role in maintaining data security.

The process initiates with the input of the AES key, securely generated and exchanged using Kyber1024 as detailed in Module 1. This key is central to encrypting large data volumes swiftly without compromising security. Subsequently, data encryption occurs where the AES key transforms plaintext into ciphertext. This transformation ensures that the data remains unintelligible to unauthorized entities, thus safeguarding against data breaches during transmission [19].

After encryption, the encrypted data is transmitted alongside the ciphertext generated during the key encapsulation phase. The ciphertext holds the encapsulated AES key, ensuring that only the intended recipient, possessing the corresponding private key, can decapsulate the shared secret and decrypt the data [20]. On the recipient's end, the decapsulation of the shared secret is performed using the private key. This critical operation, re-establishes the AES key necessary for decryption. Finally, the decryption of data is executed using the shared secret, restoring the original plaintext. This step not only recovers the data but also verifies its integrity, confirming that it remains unaltered during transit.

Algorithm 3 Signing and Verification Module

```
Generate keypair using a post-quantum digital signature scheme
Sign the input data using the private key
Transmit the data and signature
Recipient verifies the signature using sender's public key

Accept or reject the data based on verification result
if valid then
    Accept data
else
    Reject data
end if
```

C. MODULE 3: SIGNING AND VERIFICATION

Module 3 describes the digital signing and verification process using the Dilithium5 signature scheme. In addition to signing encrypted data, this same mechanism is reused to authenticate Kyber public keys during the key exchange process. The receiver verifies the sender's Kyber public key using the attached Dilithium5 signature and the sender's public key retrieved from the Ethereum smart contract. This ensures tamper-resistant and decentralized verification, preventing MITM attacks.

The Module addresses critical challenges posed by quantum computing and contributes to post-quantum secure communications, blockchain protocols, and decentralized infrastructures. Central to this mechanism is Dilithium5, a lattice-based digital signature scheme resistant to quantum attacks due to its reliance on the Module Learning With Errors (MLWE) problem [21]. This makes it a future-proof solution for secure authentication and digital identity validation. The Module addresses critical challenges posed by quantum computing and contributes to post-quantum secure communications, blockchain protocols, and decentralized infrastructures. Central to this mechanism is Dilithium5, a lattice-based digital signature scheme resistant to quantum attacks due to its reliance on the Module Learning With Errors (MLWE) problem. This makes it a future-proof solution for secure authentication and digital identity validation [22].

Initially, the Module activates the Dilithium5 signature protocol through an initialization step, creating the necessary cryptographic framework. It then generates a pair of cryptographic keys, comprising a public and private key. The private key is utilized for generating digital signatures, whereas the public key enables independent verification by recipients. This asymmetric cryptographic setup guarantees integrity and non-repudiation in digital communication.

The signing process includes data hashing, randomness generation, and lattice-based mathematical operations. The resulting signature provides cryptographic assurance of authenticity and integrity. Once the signed data is transmitted, recipients verify the signature by recalculating the expected value and comparing it with the received one. If they match, the data is accepted; otherwise, it is rejected to avoid security risks [21].

Unlike traditional signature schemes such as RSA or ECDSA, which are vulnerable to quantum attacks via Shor's algorithm, Dilithium5 maintains resilience due to its lattice-based hardness assumptions. Its performance remains strong in both classical and quantum threat models. The efficiency of Dilithium5's signature generation and verification processes also makes it ideal for high-throughput environments such as blockchain networks, financial systems, and secure messaging applications. Its compact signature structure and low computational overhead enhance its suitability for resource-constrained devices, particularly in IoT ecosystems.

The use of ECDSA is not cryptographically necessary for forward security, as Dilithium5 alone provides post-quantum security guarantees. However, its inclusion is practically necessary for compatibility with current blockchain infrastructures such as Ethereum. While Dilithium5 provides strong post-quantum digital signatures, the proposed system also includes ECDSA as a secondary signature mechanism. This dual-signature approach addresses the current limitations of blockchain platforms such as Ethereum, which rely on ECDSA for transaction validation and lack native support for quantum-safe signature verification. To bridge this gap, the system uses ECDSA alongside Dilithium5, combining both into a Merkle root that is stored and verified via smart contracts. This hybrid design ensures interoperability with legacy blockchain infrastructures while embedding post-quantum verification within the same framework. Although ECDSA introduces a known vulnerability to quantum attacks, it is necessary for backward compatibility. Once Ethereum or similar platforms evolve to natively support post-quantum signature schemes, the system can rely solely on Dilithium5 for both signing and verification.

Algorithm 4 Smart Contract Interaction Algorithm

```
Connect to the blockchain network
Retrieve user credentials securely
Load smart contract interface definition
Access deployed smart contract using its address
Prepare for contract interaction (e.g., data submission or verification)
```

V. SMART CONTRACT INTERACTION, TRANSACTION CREATION AND SENDING MODULE

A. MODULE 4: SMART CONTRACT INTERACTION

Module 4 describes the interaction with the smart contract deployed on the blockchain network.

The Module provides an essential mechanism to integrate cryptographic protocols effectively with blockchain technologies. By clearly defining interactions between a Python-based client application and a blockchain-deployed smart contract, this approach ensures secure, verifiable,

and tamper-resistant data transactions. The utilization of the Web3 library is particularly crucial, as it facilitates smooth communication between client-side applications and Ethereum-based blockchain networks, thereby effectively bridging secure data processing with decentralized ledger management.

Initially, the Module establishes a connection to the blockchain using the Web3 library, which is indispensable for blockchain-based communications. Through specific initialization commands, a Web3 instance connects to a local Ethereum environment provided by tools like Ganache. Such setups simulate realistic blockchain conditions, allowing developers and researchers to thoroughly evaluate and refine smart contract functionalities prior to their deployment on public blockchain networks [23].

Subsequently, the Module prioritizes the secure handling of sensitive information by loading essential credentials from environment variables. Critical information, including account addresses and private keys, is securely retrieved through this process. This practice significantly reduces risks associated with credential exposure, thus enhancing the overall security framework and resilience of the system.

In our implementation, the smart contract's Application Binary Interface (ABI) is loaded to enable structured interaction between the client application and the deployed contract. This step allows function calls and verification processes to be executed programmatically.

Next, retrieving the deployed smart contract's specific blockchain address is a crucial task. This unique address ensures precise interaction with the correct instance of the smart contract, which is particularly significant when multiple contracts coexist across diverse blockchain environments, including testnets and mainnets.

The final procedural step involves initializing the smart contract instance by pairing its ABI with its blockchain address. This initialization grants programmatic access to all the smart contract's functionalities, enabling secure execution of transactions, data querying, and interaction with blockchain-driven events [23].

The described Smart Contract Interaction Module profoundly influences blockchain research, secure communications, and decentralized application (dApp) development through its robust architecture and security-oriented design. Its contributions can be categorized into several key domains.

Firstly, it significantly enhances secure blockchain integration. By safeguarding sensitive credentials using environment variables and ensuring secure smart contract initialization, the Module effectively addresses prevalent security concerns, including risks associated with private key exposure and unauthorized contract interactions. Such secure integration practices are essential for building reliable blockchain-based systems.

Secondly, this Module substantially advances interoperability and standardization within the blockchain space. By leveraging standardized ABIs and the widely-adopted Web3 library, it facilitates consistent and streamlined interactions across diverse blockchain networks and client applications. This capability is vital for the creation of versatile dApps capable of operating seamlessly across multiple blockchain platforms, such as Ethereum, Binance Smart Chain, and Polygon, thereby promoting a more interconnected and interoperable blockchain ecosystem.

Additionally, the Module emphasizes data integrity and transparency, core tenets of blockchain technology. Smart contracts inherently guarantee immutability and transparency. Through secure interactions facilitated by this Module, data integrity is preserved throughout the entire transaction process [24]. Such a capability is especially valuable for critical applications like supply chain management, digital identity systems, and voting platforms, where data authenticity and traceability are paramount.

Furthermore, the Module fosters the creation of trustless digital environments, foundational to blockchain's core principles. It enables secure interactions without reliance on centralized authorities, significantly enhancing transparency, security, and resilience across distributed networks. Such decentralization is crucial for developing applications demanding high levels of trust and security assurance.

Algorithm 5 Transaction Creation and Sending Module

1: Define a unique identifier for the public key
2: Retrieve current transaction nonce for the sender
3: Construct the transaction with required parameters
4: Sign the transaction using the sender's private key
5: Broadcast the signed transaction to the blockchain
6: Record the transaction identifier for reference

B. MODULE 5: TRANSACTION CREATION AND SENDING

Module 5 describes the process of creating, signing, and sending a transaction securely on the blockchain.

The Module 5 outlines the comprehensive process of securely creating, signing, and broadcasting transactions within a blockchain network. This Module is central to decentralized systems, providing critical guarantees of data integrity, authenticity, and non-repudiation. By effectively combining cryptographic techniques with blockchain technology, it establishes a secure foundation for applications spanning decentralized finance (DeFi), digital identity management, and supply chain operations.

Initially, the Module specifies a public key identifier, referred to as the key name, uniquely distinguishing each stored public key on the blockchain. This identifier may represent specific users, devices, or other relevant application-specific entities, enabling straightforward retrieval and management within smart contracts. Consequently, the system maintains clear organization and precise identification for cryptographic keys [24].

Subsequently, the Module retrieves a nonce, a unique sequential number assigned to each blockchain transaction originating from a given address. The nonce mechanism

effectively prevents replay attacks, where adversaries attempt to maliciously resend previously transmitted transactions to compromise data or financial assets.

The next phase involves constructing the transaction through the smart contract's storePublicKey function. This step assembles a transaction intended to record encrypted data, encapsulated keys, and digital signatures onto the blockchain [25]. Essential transaction parameters such as gas limits, gas prices, sender address, and nonce are included, ensuring accurate execution and precise fee calculations within the blockchain network.

Once the transaction is assembled, the Module mandates signing it with the sender's private key. This action authenticates the transaction, establishing its validity and confirming it originated from the legitimate owner. Digital signatures reinforce non-repudiation, making it impossible for senders to deny initiating transactions. Typically, this step leverages cryptographic protocols like the Elliptic Curve Digital Signature Algorithm (ECDSA).

Following signature, the transaction is broadcasted to the blockchain network, entering the mempool. Here, it awaits confirmation through inclusion in a block, a process facilitated by network validators or miners, depending on the consensus Module employed.

Finally, the Module confirms transaction submission by displaying the transaction hash, which uniquely identifies the transaction within the blockchain. This hash provides transparency, enabling tracking of transaction status and verifying its successful inclusion within the ledger.

Module 5 substantially impacts research areas such as blockchain technology, secure communication frameworks, and cybersecurity [25]. Its robust design ensures transaction security, data integrity, and resilience to decentralized system threats.

Primarily, the Module enhances transaction security through rigorous cryptographic authentication. By employing nonces to counter replay attacks and digital signatures to affirm transaction authenticity, it significantly reduces common security vulnerabilities.

Moreover, it supports decentralization and promotes trustless interactions, core blockchain principles. By facilitating secure transactions without reliance on central authorities, the Module enhances transparency, security, and robustness across decentralized networks. This decentralization minimizes the risks of centralized control and mitigates single point of failure [26].

In terms of blockchain scalability, the Module optimizes resource utilization, effectively managing gas fees and nonces. This resource management is crucial for achieving high transaction throughput, a necessity for scalable blockchain applications like DeFi platforms, supply chain networks, and extensive IoT ecosystems.

Additionally, the Module encourages interoperability within blockchain environments. Through adherence to standardized interfaces and the strategic use of Web3 libraries, it enables seamless integration and communication across multiple blockchain platforms [26]. Such interoperability is essential for developing versatile decentralized applications (dApps) capable of operating seamlessly across diverse blockchain infrastructures.

VI. POTENTIAL APPLICATIONS OF PROPOSED FRAMEWORK

Module 3, which implements the Dilithium5 signature scheme, provides exceptional adaptability, reinforcing data authenticity and integrity across diverse sectors. In secure messaging systems, it effectively prevents unauthorized impersonation and access, ensuring message authenticity and reliability. Its application extends prominently to software integrity validation, confirming the legitimacy of software updates and downloads, thereby significantly reducing vulnerabilities to supply chain compromises and malware insertion. Within financial systems, Dilithium5 ensures robust digital transaction authentication, delivering essential non-repudiation capabilities and mitigating fraud risks. Moreover, due to its efficient computational design, the Module is particularly advantageous in IoT environments, offering secure device authentication that guards against spoofing and unauthorized interactions. It further supports secure digital governance and identity verification processes, playing a critical role in e-voting platforms, digital document signatures, and secure access to government digital services. Such extensive applicability emphasizes its importance in safeguarding digital infrastructure from current and emerging cybersecurity threats.

Module 2, which manages encryption and decryption, further extends these robust security features by emphasizing data confidentiality. It is highly effective in protecting communications within end-to-end encrypted messaging applications, offering quantum-resistant security suitable for both personal and corporate data exchanges. In the financial domain, the Module ensures the confidentiality of sensitive transactional data, effectively minimizing potential exposure to fraud and unauthorized intrusions. Additionally, the healthcare industry significantly benefits from the Module's capabilities, particularly in safeguarding electronic health records (EHRs) and facilitating compliance with rigorous data protection standards, including the Health Insurance Portability and Accountability Act (HIPAA). Its streamlined performance characteristics render it exceptionally appropriate for IoT deployments, which typically have limited computational capacities. Furthermore, the Module significantly enhances the security of governmental and military communication systems, effectively protecting sensitive and classified information against sophisticated threats.

VII. SIMULATION RESULTS

The implementation was conducted using Python 3.12.3, employing the liboqs library (version 0.11.1-dev) for Kyber1024 and Dilithium5 cryptographic operations, OpenSSL for ECDSA signatures, and AES encryption through the PyCryptodome library. Ethereum smart contract

TABLE 1. Cryptographic primitive timing (Average over 10 Iterations).

Algorithm	KeyGen Time (s)	Encrypt/Sign (s)	Decrypt/Verify (s)
AES	0.000001	0.000059	0.000012
Kyber1024	0.000022	0.000157	0.000008
Dilithium5	0.000150	0.000541	0.000790
ECDSA	0.000720	0.004698	0.006151

TABLE 2. Blockchain transaction performance metrics.

Metric	Value
Gas Usage	24,736 units
Estimated Gas Cost (ETH)	0.000049472
On-Chain Storage Requirement	0.06 KB
Transaction Delay	0.034146 seconds

TABLE 3. Key exchange performance table (seconds).

Run Number	Proposed Framework	TLS 1.3 + PQC	CECPQ2
1	0.001595	0.014076	0.000174
2	0.000105	0.000614	0.000201
3	0.000083	0.000317	0.000180
4	0.000077	0.000177	0.000176
5	0.000078	0.000176	0.000179
6	0.000077	0.000201	0.000201
7	0.000077	0.000179	0.000189
8	0.000077	0.000175	0.000174
9	0.000075	0.000175	0.000175
10	0.000076	0.000175	0.000176

interactions were managed via the web3.py library, utilizing Ganache CLI v7.9.2 as the local blockchain testing environment.

To evaluate the practicality of the proposed hybrid cryptographic scheme, we measured and compared the runtime performance of core cryptographic operations across AES, Kyber1024, Dilithium5, and ECDSA. The experiments were conducted over 10 iterations, and the results are summarized in Table 1 and visualized in Figure 6.

As shown, symmetric encryption using AES is extremely fast, with encryption and decryption times in the range of microseconds. Kyber1024 exhibits highly efficient key encapsulation and decapsulation, averaging 0.000157 s and 0.000008 s respectively, making it suitable for post-quantum key exchange in resource-constrained environments. Dilithium5, while post-quantum secure, shows slightly higher signing (0.000541 s) and verification (0.000790 s) times, though still within acceptable limits for real-world deployment.

In contrast, classical ECDSA operations are significantly slower, with signing and verification times of 0.004698 s and 0.006151 s respectively, nearly an order of magnitude higher than Dilithium5. This further emphasizes the potential performance advantages of quantum-safe digital signatures.

Additionally, Table 2 presents the blockchain-level metrics associated with storing and validating public keys via smart contracts. The total gas usage was 24,736 units, resulting in a nominal cost of 0.000049472 ETH. The total on-chain storage footprint was approximately 0.06 KB, and the transaction delay averaged 0.034146 seconds.

These results demonstrate the feasibility of integrating quantum-safe primitives into blockchain environments without introducing significant computational or economic overhead.

This section presents the simulation results of the proposed quantum-safe key exchange and signature scheme compared to two reference schemes: TLS 1.3 + PQC and CECPQ2 [27]. The performance evaluation is conducted over multiple runs, assessing key exchange, encryption/decryption, and signing/verification times. The analysis highlights the efficiency and security benefits of the proposed framework, particularly in scenarios requiring rapid and secure cryptographic operations.

The simulations were performed on a virtual system with Intel(R) Core(TM) Ultra 5 125H CPU, 4GB RAM, Ubuntu 24.04.1 LTS Operating System using an optimized cryptographic library supporting both post-quantum cryptography (PQC) and classical schemes. The key exchange performance was measured using Kyber-based hybrid encryption, while signing/verification incorporated Dilithium and ECDSA signatures. The reference algorithms followed standardized implementations within the TLS 1.3 framework and CECPQ2 [27] hybrid key exchange.

A. CPU AND MEMORY CONSUMPTION

To evaluate the system's computational resource demands, we recorded CPU and memory usage during end-to-end simulations. Over 20 runs, the mean CPU usage increase was 15.51% (95% CI: 5.02% – 26.00%), and the memory usage increase (RSS) was consistently 13.5 MB. These results (Figure 4) confirm the lightweight nature of the proposed cryptographic stack.

B. THROUGHPUT UNDER LOAD

To evaluate scalability, the system was tested under simulated load conditions using multi-threaded transaction submissions. Varying thread counts ranging from 1 to 50 were used to issue concurrent transactions to the deployed smart contract. The resulting throughput, measured in transactions per second (TPS), increased with concurrency up to a practical peak and then plateaued due to system overhead. A maximum TPS of approximately 52.92 was observed at 10 concurrent threads. Figure 3 shows how throughput evolves with increasing thread count, demonstrating that the proposed framework can support real-world blockchain-based IoT authentication scenarios under parallel load.

C. GAS COST ANALYSIS

Ethereum gas usage was benchmarked for the transaction storing the encrypted AES key, Dilithium signature, and metadata. The average gas consumption was approximately 24,736 units (0.000049 ETH), with occasional peaks reaching 28,000 units in edge cases with larger payloads. This demonstrates the gas-efficiency of our Merkle-based dual signature design.

TABLE 4. Benchmark summary (Mean ± StdDev, 95% CI).

Component	Mean (s)	StdDev	95% CI
Kyber Encapsulation	0.000648	0.000553	(0.000389, 0.000907)
Dilithium Signing	0.000596	0.000363	(0.000426, 0.000766)
Web3 Interaction	0.025177	0.004005	(0.023303, 0.027052)
Total Time	1.2894	0.1578	(1.2155, 1.3633)
CPU Usage (%)	15.51	22.42	(5.02, 26.00)
Memory Used (MB)	13.5	0	(13.5, 13.5)

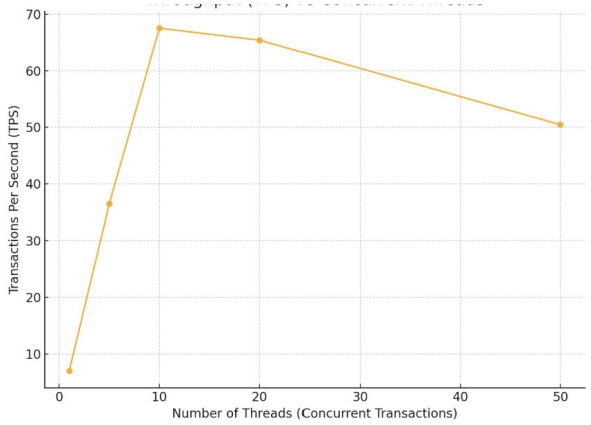


FIGURE 3. Measured throughput (TPS) under varying levels of concurrency. The system achieves optimal performance at moderate thread counts, showing saturation beyond 20 threads due to processing overhead.

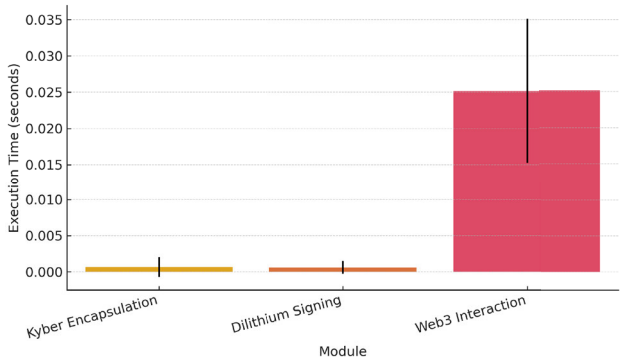


FIGURE 4. Mean runtime with 95% confidence intervals for core modules.

D. CONFIDENCE INTERVALS AND RUNTIME DISTRIBUTION

To assess statistical reliability, we computed 95% confidence intervals (CI) for runtime, CPU, and memory metrics. To visually reinforce this, Figure 4 and Figure 5 present a bar chart with error bars and a boxplot of runtime distributions, respectively. Table 4 summarizes the performance benchmarks of the core components of the proposed system. For each major operation, Kyber key encapsulation, Dilithium signature generation, and Web3-based blockchain interaction, the mean runtime, standard deviation, and 95% confidence intervals are reported across multiple iterations. The total execution time reflects the end-to-end duration for a full transaction cycle.

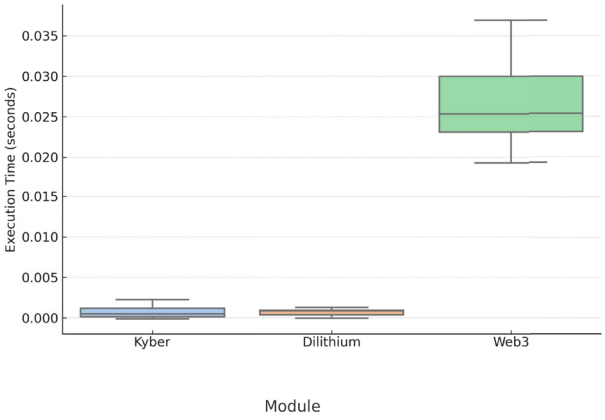


FIGURE 5. Runtime Distribution for Kyber, Dilithium, and Web3 components.

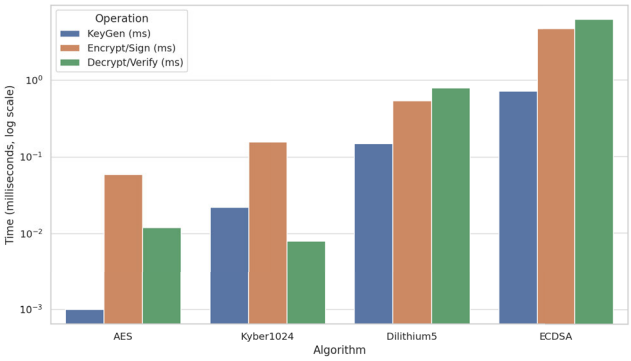


FIGURE 6. Comparison of cryptographic operation timings across AES, Kyber1024, Dilithium5, and ECDSA (log scale in milliseconds). Values averaged over 10 iterations.

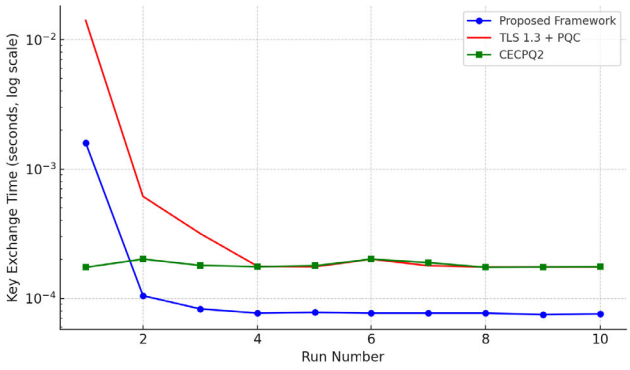


FIGURE 7. Key exchange performance comparison.

Fig. 7 and Table 3 illustrates the key exchange performance over ten runs for the proposed framework, TLS 1.3 + PQC, and CECPQ2. The proposed framework consistently demonstrates lower latency compared to TLS 1.3 + PQC, which exhibits a significant delay in the initial run before stabilizing. CECPQ2 maintains a relatively stable but slightly higher execution time than the proposed approach.

The superior performance of the proposed framework is attributed to its optimized key encapsulation mechanism,

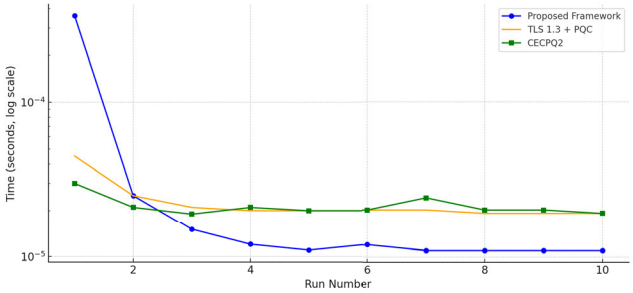


FIGURE 8. Encryption/ Decryption performance comparison.

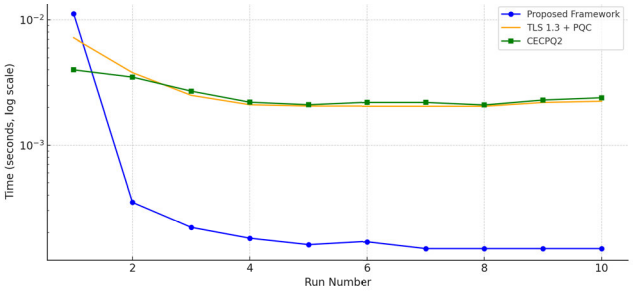


FIGURE 9. Signing/ Verification performance comparison.

which reduces computational overhead. Unlike TLS 1.3 + PQC, which suffers from high initial handshake delays due to large key sizes and complex hybrid operations, the proposed method efficiently handles key exchange while maintaining robustness against quantum threats [27].

Fig. 8 illustrates the comparison of encryption and decryption time among the three algorithms. Although TLS 1.3 + PQC and CECPQ2 exhibit relatively consistent performance across all runs, the proposed framework initially records a higher delay (0.00035 seconds) during the first execution. This one-time latency is attributed to initialization overhead, such as cryptographic context setup and memory allocation for symmetric key expansion.

Despite this early peak, the proposed framework rapidly converges to a lower and more consistent latency, outperforming both TLS 1.3 + PQC and CECPQ2 in all subsequent runs. This shows the advantage of the proposed scheme in repeated and sustained use scenarios, where initialization costs are paid off. The trade-off between a slight one-time setup delay and long-term superior performance makes the proposed framework a practical and efficient choice for secure communications.

One of the key advantages of the proposed encryption scheme is its streamlined integration of symmetric keys that minimizes transmission overhead. By leveraging hybrid encryption with post-quantum security guarantees, it achieves a better trade-off between security and efficiency. This is particularly critical for real-time applications where rapid encryption and decryption are necessary to maintain secure communication without noticeable delays [28].

Fig. 9 presents the signing and verification performance comparison. Initially, the proposed framework exhibits a peak delay (0.011 seconds), higher than both CECPQ2 and TLS 1.3 + PQC. However, after the first two runs, the proposed framework stabilizes and significantly outperforms the reference schemes.

This initial peak is primarily attributed to one-time setup operations, such as cryptographic parameter initialization and key structure allocation, which occur during the first execution. These are common overheads in cryptographic implementations and do not reflect the steady-state performance.

After the initial run, the proposed framework rapidly stabilizes, demonstrating the lowest execution times across all subsequent runs. This highlights its efficiency in practical, repeated-use scenarios where setup cost is amortized. Despite this minor one-time latency, the consistent superiority in performance clearly underscores the practicality and computational advantage of the proposed approach in real-world deployments.

This efficiency gain is largely due to the optimized implementation of the Dilithium signature scheme in the proposed approach. Unlike Falcon-based schemes, which require complex floating-point operations, Dilithium utilizes integer-based computations, leading to faster execution in constrained environments. Moreover, the signature verification time remains consistently lower, ensuring efficient authentication in secure communications.

The proposed framework outperforms TLS 1.3 + PQC across all categories, particularly in key exchange and signing operations. While CECPQ2 maintains competitive performance, it does not provide the same level of post-quantum security as the proposed approach [28].

E. SECURITY ANALYSIS

Beyond performance, security considerations play a crucial role in evaluating cryptographic protocols. The proposed framework stands out due to its integration of lattice-based cryptographic primitives, which offer superior resilience against quantum attacks. In contrast to CECPQ2, which continues to depend heavily on traditional elliptic-curve cryptography (ECC), the proposed methodology effectively addresses the long-term security requirements posed by quantum computing threats. Although TLS 1.3 augmented with post-quantum cryptography (PQC) integrates quantum-resistant algorithms, it frequently encounters challenges such as large key sizes and complex hybrid operations, impacting efficiency [29]. Our proposed method substantially reduces these issues by employing the Kyber encapsulation scheme, an advanced key-exchange technique that significantly lowers data transmission overhead while upholding robust security standards. This strategy is particularly advantageous in computationally constrained environments, allowing for efficient key exchanges with minimal latency.

Furthermore, for digital signature applications, our method adopts the Dilithium algorithm, recognized for effectively balancing security and computational performance. In contrast to Falcon, which demands intricate floating-point

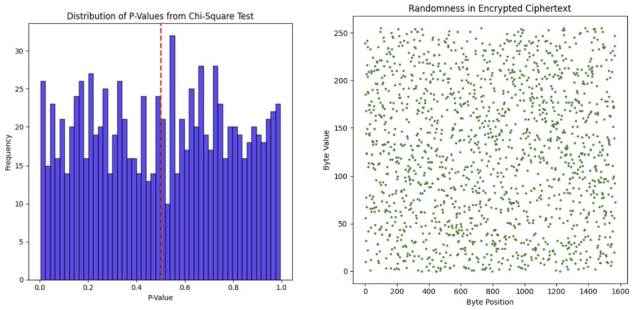


FIGURE 10. Encryption randomness and security strength of the proposed framework.

arithmetic, Dilithium relies entirely on integer-based operations. This characteristic greatly enhances practicality for both hardware and software implementations [29], especially beneficial in resource-limited contexts such as IoT devices, where computational efficiency is critically important. The streamlined design of Dilithium's signing and verification processes contributes to accelerated authentication performance relative to TLS 1.3 with PQC enhancements and CECPQ2 [28]. Overall, this results in a cryptographic system that not only provides robust security against quantum threats but is also optimized for practical, real-world implementation, marking it as an ideal solution for advanced cybersecurity applications.

According to Figure 10, the results from the statistical analysis of the encrypted ciphertext provide strong evidence for the security and robustness of our encryption scheme. The histogram of p-values from the Chi-Square test demonstrates that the ciphertext is statistically indistinguishable from random noise, ensuring that adversaries cannot extract meaningful patterns. The p-values are uniformly distributed across the range of 0 to 1, with no observable bias toward lower values. A non-random encryption scheme would exhibit a skewed p-value distribution, typically clustered toward 0, indicating a deviation from uniform randomness. The fact that our encryption scheme maintains p-values around 0.5 signifies a well-distributed randomness pattern, reinforcing its security against statistical attacks. This property is critical for preventing adversaries from using frequency analysis techniques to infer plaintext characteristics from ciphertext distributions.

The scatter plot of encrypted byte values further validates the randomness of our ciphertext by showing an even distribution across all byte positions. If an encryption scheme were vulnerable, one would expect to observe clusters or structured patterns within the scatter plot, which could reveal information about the plaintext structure. However, in our results, the byte values are uniformly dispersed, demonstrating that our encryption scheme effectively conceals any underlying patterns. This randomness is essential in cryptographic applications, ensuring that even if an attacker captures multiple encrypted messages, they remain indistinguishable from one another. Such unpredictability strengthens resistance against chosen-plaintext and known-plaintext attacks, which rely on identifying correlations between ciphertexts and their corresponding plaintexts.

The implications of these findings are significant in the context of post-quantum cryptography and secure communications. Traditional encryption schemes, such as RSA and ECC, are vulnerable to quantum attacks, where adversaries leveraging Shor's algorithm can efficiently break their security assumptions. Our encryption mechanism, based on Kyber1024, ensures resilience against both classical and quantum adversaries. By achieving statistical randomness, our approach mitigates the risk of quantum-enabled cryptanalysis techniques that might exploit predictable encryption patterns [32].

Beyond its theoretical security benefits, our encryption approach has practical applications in industries requiring high-assurance cryptographic security, such as financial transactions, government communications, and blockchain-based data protection. The ability to generate truly random ciphertext ensures compliance with stringent security standards. Additionally, our results indicate that the proposed encryption scheme is well-suited for real-world deployment in secure messaging systems, VPNs, and cloud storage encryption. As data privacy concerns grow globally, adopting cryptographic methods that guarantee indistinguishability and resilience against advanced adversarial models is crucial. Our encryption scheme offers a forward-looking solution to these challenges, ensuring long-term security and privacy for sensitive communications in a rapidly evolving cybersecurity landscape.

The use of Dilithium5 and Kyber1024 follows NIST's standardized forms ML-DSA and ML-KEM (FIPS 203/204) [14], [15], which include guidelines for side-channel resistance. Although our current implementation uses pre-standard versions, the structure is fully compatible with hardened implementations. Future work includes migrating to side-channel hardened variants from libraries such as PQClean and liboqs, which utilize masking and constant-time operations to mitigate timing and power analysis risks.

As per Figure 11, the results obtained from the signature forgery test provide strong evidence for the robustness of the Dilithium signature scheme. The left-side histogram in the visualization represents the outcome of 1000 forgery attempts, with a clear indication that every attempt has resulted in failure. This suggests that the scheme effectively resists forgery attacks even when adversaries attempt to modify signatures [30]. A compromised digital signature scheme would have shown a portion of successful forgery attempts, represented by a nonzero count in the "Forgery Succeeded" category. However, the complete absence of successful forgeries in our results indicates that Dilithium maintains the integrity and authenticity of digital signatures under adversarial conditions. Such a level of security is essential in applications where trust and non-repudiation are required, such as secure messaging, blockchain authentication, and financial transactions.

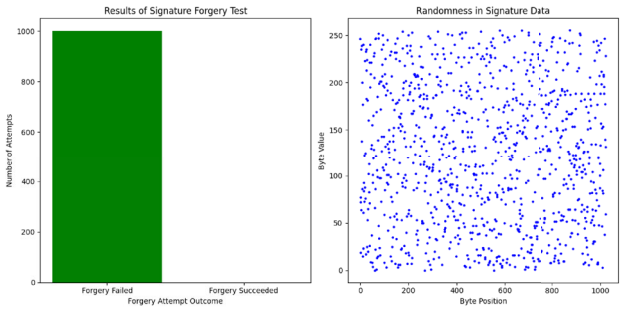


FIGURE 11. Analysis of signature forgery.

The scatter plot on the right provides further validation of the security of the signature scheme. The distribution of byte values across different positions in the generated signatures appears random and uniformly spread, demonstrating strong entropy in the cryptographic signing process. If the signature generation process had predictable structures or patterns, attackers could leverage statistical analysis to infer potential weaknesses, thereby increasing the risk of forgery [30]. However, the random distribution seen in our results indicates that signature data remains resistant to structured pattern recognition. This randomness ensures that each generated signature is unique and cannot be correlated with previously observed signatures, further strengthening the security of the scheme.

One of the significant advantages of the Dilithium signature scheme is its resistance to both classical and quantum attacks. Traditional cryptographic signature schemes such as RSA and ECDSA are vulnerable to quantum adversaries due to Shor's algorithm, which can efficiently break their underlying security assumptions [33]. However, Dilithium is based on lattice-based cryptography, a post-quantum secure approach that remains resilient even against quantum-powered attacks. The resistance to forgery observed in our tests further reinforces this claim, making it a suitable candidate for next-generation cryptographic applications. Industries such as government communications, secure software updates, and cloud security can benefit from adopting this technology to ensure the authenticity and security of their digital transactions.

Hybrid schemes must mitigate downgrade attacks where adversaries substitute the quantum-safe component with a weaker one. In our scheme, Dilithium signatures authenticate Kyber public keys prior to key exchange, binding them cryptographically to prevent replacement or misuse. Additionally, Merkle roots incorporating both ECDSA and Dilithium signatures prevent selective signature tampering and substitution attacks.

The results of this study demonstrate that the Dilithium signature scheme is a highly reliable cryptographic solution for securing digital identities. The complete failure of forgery attempts and the observed randomness in signature byte distributions indicate that an attacker cannot forge valid signatures or extract meaningful insights from signature patterns. This makes Dilithium an excellent choice for securing digital ecosystems that require long-term security guarantees. With the increasing threat of quantum computing, transitioning to post-quantum cryptographic algorithms such as Dilithium is a proactive step toward safeguarding sensitive data and communications.

Hybrid schemes must mitigate downgrade attacks where adversaries substitute the quantum-safe component with a weaker one. In our scheme, Dilithium signatures authenticate Kyber public keys prior to key exchange, binding them cryptographically to prevent replacement or misuse. Additionally, Merkle roots incorporating both ECDSA and Dilithium signatures prevent selective signature tampering and substitution attacks. The proposed scheme follows NIST's recommendations outlined in NISTIR 8413 for hybrid transition mechanisms [31]. Specifically, the system employs a modular architecture where classical components (e.g., ECDSA) can be replaced by quantum-safe primitives (e.g., Dilithium5) as platforms mature. Backward compatibility is maintained for Ethereum through dual signature Merkle-root-based proofs. The architecture is forward-compatible with rollups and forks enabling PQC-native consensus. These findings provide a strong foundation for advocating the adoption of this scheme in real-world applications requiring unforgeable and quantum-safe digital signatures.

VIII. CONCLUSION AND FUTURE WORK

A. LIMITATIONS AND OPEN CHALLENGES

While the proposed hybrid cryptographic framework offers robust post-quantum security and efficient blockchain integration, several limitations must be acknowledged.

1) ECDSA Vulnerability to Quantum Attacks: The use of ECDSA, though necessary for compatibility with current Ethereum infrastructures, introduces a known vulnerability. Shor's algorithm could compromise ECDSA if scalable quantum computers emerge. The system addresses this through dual-signature Merkle root verification but remains reliant on ECDSA for transaction authorization.

2) Signature Size and Gas Costs: Post-quantum signature schemes such as Dilithium5 typically produce larger signatures (2–3 KB) than classical schemes. Although mitigated using Merkle roots and hybrid design, large signature payloads may still increase gas costs, affecting on-chain storage efficiency.

3) Verification Overhead: While Dilithium5 performs efficiently in simulations, its verification latency is higher than ECDSA. In large-scale blockchain environments with high transaction volumes, this overhead might affect throughput unless Ethereum or similar platforms introduce PQC-optimized verification primitives.

4) Deployment Readiness of Ethereum: The full migration to pure post-quantum signatures (e.g., removing ECDSA entirely) requires substantial changes to the Ethereum protocol. Our reliance on backward-compatible

dual signature systems highlights the transitional nature of the current solution.

5) Library and Standard Dependency: Current implementation uses pre-standardized versions of Kyber and Dilithium. While structurally compatible with ML-KEM and ML-DSA, full transition to side-channel hardened libraries such as PQClean is part of ongoing work.

6) Integration Complexity: Incorporating zero-knowledge proofs (zk-SNARKs), multi-party computation (MPC), or cross-chain post-quantum interoperability introduces design and performance complexities that require further research and optimization.

The proposed mechanism successfully integrates quantum-safe cryptographic algorithms with blockchain technology to enhance security in decentralized applications. By using Kyber for key exchange, authenticated via Dilithium5 signatures verified using keys stored on Ethereum smart contracts to prevent MITM attacks, along with ECDSA for blockchain compatibility and AES encryption for data confidentiality, it provides a secure and verifiable framework resistant to both classical and quantum threats [32], [33]. The use of Ethereum smart contracts ensures tamper-proof storage and verification.

As for the benefits of integrating Quantum-Safe Cryptography with blockchain, the hybrid cryptographic approach mitigates the risks posed by quantum computers. The integration of Merkle root verification ensures data integrity in an immutable ledger. Supporting both classical (ECDSA) and quantum-safe (Dilithium) signatures allows for seamless transition and compatibility with existing blockchain ecosystems. The use of smart contracts for verification allows decentralized and scalable security solutions. Future implementations will migrate fully to ML-KEM and ML-DSA, ensuring strict alignment with NIST's finalized specifications and incorporating state-of-the-art resistance against implementation-level attacks such as SCAs and FAs.

The future of quantum-safe cryptographic frameworks for blockchain applications involves several key areas of advancement. Enhancing scalability by implementing more advanced quantum-safe schemes such as Multi-Layer Digital Signature Algorithm (ML-DSA) and Modular and Adaptive cryptography for post-quantum security (MAYO) can significantly increase cryptographic robustness [34]. Optimizing transaction costs through reduced computational overhead associated with larger key sizes and signature verification remains a critical challenge. Additionally, integrating Zero-Knowledge Proofs (ZKPs) can provide enhanced privacy-preserving features, allowing for transaction verification without exposing sensitive data. Implementing Zero-Knowledge Succinct Non-Interactive Argument of Knowledge (zk-SNARKs) and Zero-Knowledge Scalable Transparent Argument of Knowledge (zk-STARKs) will optimize proof verification on Ethereum while maintaining security [35], [36]. Expanding into Multi-Party Computation (MPC) will enable secure collaborative key management, leveraging threshold cryptography techniques to distribute trust among multiple parties. Furthermore, ensuring cross-chain interoperability by integrating quantum-safe blockchain security with interoperability protocols will facilitate secure transactions across multiple blockchains [37]. Developing atomic swaps and interledger communication using quantum-resistant cryptography can further strengthen decentralized security. Overall, the presented hybrid cryptographic framework offers a robust solution for securing blockchain transactions against quantum threats [35], [36]. Future advancements in scalability, privacy through ZKPs, MPC for secure computation [38], and cross-chain interoperability will continue to enhance its real-world applicability.

Although our scheme employs ECDSA alongside Dilithium5 to ensure backward compatibility and verifiability through standard Ethereum smart contracts, this approach is inherently transitional. For future-proof blockchain transaction validation, we propose the integration of pure PQ-DSA at the consensus layer, where only quantum-safe signatures (e.g., Dilithium, SPHINCS+) are used to sign and authorize transactions. To mitigate these effects, compression techniques (e.g., zk-SNARKs) and off-chain verification models could be investigated. Another promising direction includes the design of post-quantum Layer-2 rollups or custom blockchain protocols that natively support PQ-signature formats. As part of future work, we aim to experiment with replacing ECDSA entirely using emerging Ethereum Improvement Proposals (EIPs) or private Ethereum forks capable of handling native Dilithium-based transactions.

In contrast to existing PQC-blockchain integrations such as those proposed by Castiglione et al. [6] and Fernández-Caramés and Fraga-Lamas [5], our work provides a fully realized implementation and performance evaluation. We also respond to practical concerns raised by Holmes [7] regarding transaction cost and signature scalability by utilizing Merkle root–based signature proofs and hybrid encryption. These design choices make the framework not only theoretically sound but also suitable for real-world blockchain applications including smart contracts, digital identity, and IoT device authentication.

B. FUTURE WORK

To further address Ethereum's known transaction throughput bottlenecks and high gas fees, future work will investigate integrating Layer 2 scaling solutions, such as Optimistic Rollups or zkRollups, to reduce gas costs associated with large PQ signature sizes.

A future direction involves evaluating integration with decentralized KMS architectures or trusted cloud-based KMS providers (e.g., AWS KMS, Azure Key Vault) to securely manage and rotate PQ public/private keys without compromising decentralization or transparency.

It is important to note that our contribution lies not in inventing new cryptographic primitives, but in designing a robust framework that effectively integrates existing post-quantum algorithms with Ethereum-based smart contract mechanisms, offering tamper-proof verification, effi-

cient encryption, and interoperability with current blockchain ecosystems.

REFERENCES

[1] J. P. Mattsson, B. Smeets, and E. Thormarker, "Quantum-resistant cryptography," 2021, *arXiv:2112.00399*.

[2] D. Chawla and P. S. Mehra, "A roadmap from classical cryptography to post-quantum resistant cryptography for 5G-enabled IoT: Challenges, opportunities and solutions," *Internet Things*, vol. 24, Dec. 2023, Art. no. 100950.

[3] B. Senapati and B. S. Rawal, "Quantum communication with RLP quantum resistant cryptography in industrial manufacturing," *Cyber Secur. Appl.*, vol. 1, Dec. 2023, Art. no. 100019.

[4] O. S. Althobaiti and M. Dohler, "Quantum-resistant cryptography for the Internet of Things based on location-based lattices," *IEEE Access*, vol. 9, pp. 133185–133203, 2021.

[5] T. M. Fernández-Caramès and P. Fraga-Lamas, "Towards post-quantum blockchain: A review on blockchain cryptography resistant to quantum computing attacks," *IEEE Access*, vol. 8, pp. 21091–21116, 2020, doi: 10.1109/ACCESS.2020.2968985.

[6] A. Castiglione, J. G. Esposito, V. Loia, M. Nappi, C. Pero, and M. Polsinelli, "Integrating post-quantum cryptography and blockchain to secure low-cost IoT devices," *IEEE Trans. Ind. Informat.*, vol. 21, no. 2, pp. 1674–1683, Feb. 2025, doi: 10.1109/TII.2024.3485796.

[7] S. A. Holmes, "Impact of post-quantum signatures on blockchain and DLT systems," in *Proc. 5st Workshop Distrib. Ledger Technol. (DLT)*, vol. 3460, 2023. [Online]. Available: https://ceur-ws.org/Vol-3460/

[8] N. A. Ismail, S. A. Khadra, G. M. Attiya, and S. E. S. E. Abdulrahman, "Optimizing SIKE for blockchain-based IoT ecosystems with resource constraints," *J. Supercomput.*, vol. 81, no. 3, p. 463, Feb. 2025, doi: 10.1007/s11227-024-06906-z.

[9] Y. Zhang, P. Duan, C. Li, H. Zhang, and H. Ahmad, "Preserving privacy of Internet of Things network with certificateless ring signature," *Sensors*, vol. 25, no. 5, p. 1321, Feb. 2025, doi: 10.3390/s25051321.

[10] K. K. Singamaneni and G. Muhammad, "A novel integrated quantum-resistant cryptography for secure scientific data exchange in ad hoc networks," *Ad Hoc Netw.*, vol. 164, Nov. 2024, Art. no. 103607.

[11] D. Joseph, R. Misoczki, M. Manzano, J. Tricot, F. D. Pinuaga, O. Lacombe, S. Leichenauer, J. Hidary, P. Venables, and R. Hansen, "Transitioning organizations to post-quantum cryptography," *Nature*, vol. 605, no. 7909, pp. 237–243, May 2022.

[12] M. Allende, D. L. León, S. Cerón, A. Pareja, E. Pacheco, A. Leal, M. Da Silva, A. Pardo, D. Jones, D. J. Worrall, B. Merriman, J. Gilmore, N. Kitchener, and S. E. Venegas-Andraca, "Quantum-resistance in blockchain networks," *Sci. Rep.*, vol. 13, no. 1, p. 5664, Apr. 2023.

[13] C. Rubio García, S. Rommel, S. Takarabt, J. J. Vegas Olmos, S. Guilley, P. Nguyen, and I. Tafur Monroy, "Quantum-resistant transport layer security," *Comput. Commun.*, vol. 213, pp. 345–358, Jan. 2024.

[14] *Module-Lattice-Based Key-Encapsulation Mechanism Standard (ML-KEM)*, National Institute of Standard FIPS 203, National Institute of Standards and Technology, 2024, doi: 10.6028/NIST.FIPS.203.

[15] *Module-Lattice-Based Digital Signature Standard (ML-DSA)*, Standard FIPS 204, National Institute of Standards and Technology, 2024, doi: 10.6028/NIST.FIPS.204.

[16] T. M. Fernández-Caramés, "From pre-quantum to post-quantum IoT security: A survey on quantum-resistant cryptosystems for the Internet of Things," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6457–6480, Jul. 2020.

[17] S. He, H. Li, F. Li, and R. Ma, "A lightweight hardware implementation of CRYSTALS-kyber," *J. Inf. Intell.*, vol. 2, no. 2, pp. 167–176, Mar. 2024.

[18] Y. Yang, L. Wu, X. Zhang, and M. Chinbat, "Power analysis on hardware implementation of CRYSTALS-kyber," in *Proc. IEEE 18th Int. Conf. Anti-Counterfeiting, Secur., Identificat. (ASID)*, Nov. 2024, pp. 1–5.

[19] M. Chinbat, L. Wu, X. Zhang, A. Batsukh, Y. Yang, and L. Wu, "Evaluating side-channel attack vulnerabilities in post-quantum CRYSTALS-kyber hardware based on simple power analysis," in *Proc. IEEE 17th Int. Conf. Anti-Counterfeiting, Secur., Identificat. (ASID)*, Dec. 2023, pp. 46–49.

[20] L. Wan, F. Zheng, F. Guang, R. Wei, L. Gao, Y. Wang, J. Lin, and J. Dong, "A novel high-performance implementation of CRYSTALS-kyber with AI accelerator," in *Proc. Eur. Symp. Res. Comput. Secur.*, in Lecture Notes in Computer Science, vol. 13556, V. Atluri, R. Di Pietro, C. D. Jensen, and W. Meng, Eds., Springer, 2022, pp. 514–534.

[21] P. Ren, X. Gu, and Z. Wang, "Efficient module learning with errors-based post-quantum password-authenticated key exchange," *IET Inf. Secur.*, vol. 17, no. 1, pp. 3–17, Jan. 2023.

[22] S. Islam, K. Mus, R. Singh, P. Schaumont, and B. Sunar, "Signature correction attack on dilithium signature scheme," in *Proc. IEEE 7th Eur. Symp. Secur. Privacy (EuroS&P)*, Jun. 2022, pp. 647–663.

[23] H. Taherdoost, "Smart contracts in blockchain technology: A critical review," *Information*, vol. 14, no. 2, p. 117, Feb. 2023.

[24] S. N. Khan, F. Loukil, C. Ghedira-Guegan, E. Benkhelifa, and A. Bani-Hani, "Blockchain smart contracts: Applications, challenges, and future trends," *Peer-Peer Netw. Appl.*, vol. 14, no. 5, pp. 2901–2925, Sep. 2021.

[25] F. Ayotunde Alaba, H. Adewale Sulaimon, M. Ifeyinwa Marisa, and O. Najeem, "Smart contracts security application and challenges: A review," *Cloud Comput. Data Sci.*, pp. 15–41, Sep. 2023.

[26] M. Faheem, H. Kuusniemi, B. Eltahawy, M. S. Bhutta, and B. Raza, "A lightweight smart contracts framework for blockchain-based secure communication in smart grid applications," *IET Gener., Transmiss. Distrib.*, vol. 18, no. 3, pp. 625–638, Feb. 2024.

[27] D. J. Bernstein, B. B. Brumley, M.-S. Chen, and N. Tuveri, "OpenSSLNTRU: Faster post-quantum TLS key exchange," in *Proc. 31st USENIX Secur. Symp. (USENIX Secur.)*, Aug. 2021, pp. 845–862.

[28] D. Marchsreiter and J. Sepúlveda, "Hybrid post-quantum enhanced TLS 1.3 on embedded devices," in *Proc. 25th Euromicro Conf. Digit. Syst. Design (DSD)*, 2022, pp. 905–912.

[29] D. Marchsreiter and J. Sepúlveda, "A PQC and QKD hybridization for quantum-secure communications," in *Proc. 26th Euromicro Conf. Digit. Syst. Design (DSD)*, 2023, pp. 545–552.

[30] S. Shen, H. Yang, W. Dai, H. Zhang, Z. Liu, and Y. Zhao, "High-throughput GPU implementation of dilithium post-quantum digital signature," *IEEE Trans. Parallel Distrib. Syst.*, vol. 35, no. 11, pp. 1964–1976, Nov. 2024.

[31] *Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process*, Standard 8413, National Institute of Standards and Technology, 2022, doi: 10.6028/NIST.IR.8413.

[32] V. Maram and K. Xagawa, "Post-quantum anonymity of kyber," in *Proc. IACR Int. Conf. Public-Key Cryptogr.*, Cham, Switzerland, A. Boldyreva and V. Kolesnikov, Eds., Springer, 2023, pp. 3–35.

[33] A. C. H. Chen, "Post-quantum cryptography X.509 certificate," in *Proc. Int. Conf. Smart Syst. Appl. Electr. Sci. (ICSSES)*, May 2024, pp. 1–6.

[34] R. Shajahan, K. Jain, and P. Krishnan, "A survey on NIST 3rd round post quantum digital signature algorithms," in *Proc. 5th Int. Conf. Mobile Comput. Sustain. Informat. (ICMCSI)*, Jan. 2024, pp. 132–140.

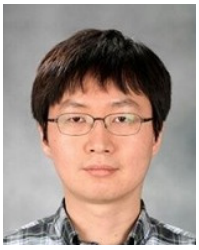
[35] L. Zhou, A. Diro, A. Saini, S. Kaisar, and P. C. Hiep, "Leveraging zero knowledge proofs for blockchain-based identity sharing: A survey of advancements, challenges and opportunities," *J. Inf. Secur. Appl.*, vol. 80, Feb. 2024, Art. no. 103678.

[36] O. Kuznetsov, A. Rusnak, A. Yezhov, D. Kanonik, K. Kuznetsova, and S. Karashchuk, "Enhanced security and efficiency in blockchain with aggregated zero-knowledge proof mechanisms," *IEEE Access*, vol. 12, pp. 49228–49248, 2024.

[37] J. A. Khan, W. Wang, and K. Ozbay, "BELIEVE: Privacy-aware secure multi-party computation for real-time connected and autonomous vehicles and micro-mobility data validation using blockchain—A study on New York city data," *Transp. Res. Rec.*, vol. 2678, no. 3, pp. 410–421, Mar. 2024.

[38] I. Zhou, F. Tofigh, M. Piccardi, M. Abolhasan, D. Franklin, and J. Lipman, "Secure multi-party computation for machine learning: A survey," *IEEE Access*, vol. 12, pp. 53881–53899, 2024.

PERERA K. MADUNI received the B.S. degree in computer security from the University of Plymouth, Plymouth, U.K., in 2021. She is currently pursuing the master's degree in computer engineering with Dong-A University, Busan, South Korea. Her current research interests include advanced cryptography, quantum resistant cryptography, digital forensics, and network security.



ILMU BYUN (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical and electronic engineering from Yonsei University, Seoul, South Korea, in 2005, 2007, and 2013, respectively. From January 2013 to November 2017, he was a Senior Research Engineer with the Advanced Standard Research and Development Laboratory, LG Electronics. Since November 2017, he has been with Korea Railroad Research Institute (KRRI). His current research interests include private 5G networks for railways, and integrated sensing and communications.

JEONGIL SEO (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electronics engineering from Kyungpook National University, Daegu, Republic of Korea, in 1994, 1996, and 2005, respectively. From 1998 to 2000, he was a member of Engineering Staff at LG Semicon, Cheongju, South Korea. From 2000 to 2022, he was the Director of the Immersive Media Research Section, Electronics and Telecommunication Research Institute (ETRI), Daejeon, South Korea. He is currently an Associate Professor with Dong-A University, Busan, South Korea. His current research interests include audio and video coding, immersive media, and computer vision.

KYEONGJUN KO (Member, IEEE) received the B.S. and Ph.D. degrees from the Department of Electrical Engineering, Seoul National University, Seoul, South Korea, in 2006 and 2012, respectively. He was a Postdoctoral Researcher with the Wireless Signal Processing Laboratory, Seoul National University, from 2012 to 2013. Then, he was a Senior Researcher with Korea Railroad Research Institute, from September 2013 to August 2023. He joined as an Assistant Professor with the Department of Computer Engineering, Dong-A University, in September 2023. Since March 2025, he has been an Assistant Professor with the School of Electronics and Electrical Engineering, Hongik University. His current research interests include low complexity deep neural networks, image processing, and automatic driving systems.