Lattice-Based Standardized Cross-Chain Payment Protocol for Post-Quantum IoT Ecosystems

Bing Zhang[1,2], Guijuan Wang[1,2,*], Yubing Han[1,2],
Xiang Tian[1,2], Chuangen Gao[1,2], Ningning Liu[3]

{bingzhang0925@163.com, guijuan_wang@126.com, cauhanbing@163.com,
tianx@qlu.edu.cn, gaochuangen@163.com, 92033@jssvc.edu.cn}

[1]Key Laboratory of Computing Power Network and Information Security, Ministry of Education,
Shandong Computer Science Center, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250353, China

[2]Shandong Provincial Key Laboratory of Industrial Network and Information System Security,
Shandong Fundamental Research Center for Computer Science, Jinan 250353, China

[3]School of Business, Suzhou Polytechnic University, Suzhou 215104, China

* *Corresponding author*

Abstract. The development of the Internet of Things (IoT) ecosystem confronts two major obstacles: interoperability barriers caused by the coexistence of multiple blockchains and long-term security threats from quantum computing to classical cryptographic systems. Existing cross-chain mechanisms generally involve high overhead, long latency, and reliance on non–post-quantum primitives, making them impractical for constrained IoT devices. To overcome these issues, this paper presents a lattice-based standardized cross-chain payment protocol designed for post-quantum security and lightweight deployment. The protocol employs a provably secure lattice signature scheme and a lightweight interoperability framework to automate cross-chain coordination. Keyword-driven embedded smart contracts ensure atomic transaction execution, while integrated IoT communication protocols improve data transmission efficiency. Security analysis proves that the scheme satisfies essential post-quantum security properties, and experimental results show significant improvements in transaction latency, computational overhead, and scalability. The proposed protocol offers a secure and efficient solution for building the next generation of quantum-resistant IoT value networks.

Keywords: lattice; cross-chain payment; IoT; post-quantum cryptography; standardization; embedded smart contract

IIKI 2025, December 18-21, Chengdu, People's Republic of China

DOI 10.4108/eai.18-12-2025.2365255

1 Introduction

Driven by the rapid growth of Internet of Things (IoT) devices and significant advancements in blockchain technology[1], building a secure, efficient, and decentralized IoT value exchange network has become a shared vision for both academia and industry. In such a paradigm, smart devices are no longer simple data collection nodes but autonomous economic agents capable of performing value exchange. By establishing trust through blockchain and achieving automated collaboration via smart contracts, it becomes possible to realize a fully decentralized IoT economy[2, 3].

However, the realization of this vision faces multiple technical and practical challenges. The inherent resource constraints of IoT devices stand in stark contrast to the high computational and storage overhead of mainstream blockchain operations, which significantly restricts large-scale deployment. More critically, when IoT devices operate across heterogeneous blockchain ecosystems, enabling secure and efficient cross-chain interaction becomes a key bottleneck. The advancement of quantum computing undermines the security of traditional cryptographic systems, particularly elliptic curve cryptography (ECC), which necessitates the urgent adoption of quantum-resistant solutions for upcoming IoT infrastructures.

To address the dual challenges of resource limitation and cross-chain interoperability, several studies have explored lightweight blockchain frameworks. For example, Zhang et al. [4] proposed a keyword-embedded smart contract model that integrates contract deployment into single-call transactions, effectively reducing latency and resource consumption while establishing a keyword-matching-based cross-chain interoperability framework. Although promising, this design still suffers from two key limitations: (i) its reliance on classical ECC renders it insecure against quantum adversaries, and (ii) its customized keyword-based communication lacks a unified standard, hindering scalability and interoperability across platforms. Similarly, mainstream cross-chain approaches such as sidechains[5] and hash time-locked contracts (HTLCs) [6] are also ill-suited for IoT scenarios due to high implementation complexity, long response delays, and the use of non–post-quantum cryptographic primitives.

In response to these issues, this work presents a standardized cross-chain payment protocol built upon lattice-based cryptography, aimed at providing a post-quantum-secure foundation for IoT value exchange. The proposed protocol integrates lattice-based cryptography with a lightweight interoperability framework, enabling secure, scalable, and efficient cross-chain transactions for resource-constrained IoT devices. Specifically, we design a lattice-signature-based cross-chain payment mechanism that achieves seamless interoperability across heterogeneous blockchains through standardized message formats while maintaining quantum resistance. Building upon the efficiency of keyword-embedded smart contracts, our scheme ensures low resource overhead and high communication efficiency. Through comprehensive security analysis and performance evaluation, this work substantiates the protocol's effectiveness and feasibility in practical IoT settings.

2 Preliminaries

2.1 Learning With Errors

At its core, the Learning With Errors (LWE) problem involves distinguishing samples of the form $(a_i, b_i = \langle a_i, s \rangle + e_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ from truly random pairs, with s being a secret and e_i a small error. This problem is conjectured to be hard for quantum computers [7], forming the cornerstone of post-quantum cryptography and underpinning the security of schemes like Dilithium.

2.2 Dilithium Signature Scheme

The security of the Dilithium signature scheme, a lattice-based construction, is founded on the hardness of the LWE and Ring-LWE problems [8, 9, 10, 11]. It operates over the polynomial ring

$$R_q = \mathbb{Z}_q[X]/(X^n + 1), \tag{1}$$

and uses bounded polynomial vectors $(\mathbf{s}_1, \mathbf{s}_2)$ as secret keys. SHAKE-256 is employed for hashing and key derivation. Dilithium consists of three algorithms: KeyGen, Sign, and Verify. As a NIST PQC finalist, Dilithium provides clear parameter sets and efficient performance, making it suitable for lightweight cross-chain payment protocols requiring post-quantum security.

2.3 Embedded Smart Contracts

Optimization efforts for smart contracts have traditionally targeted execution efficiency, for example by eliminating redundant operations [12] or removing unnecessary components [13]. Building on this line of work, Su et al. [14] proposed an embedded contract paradigm in which deployment and invocation are consolidated into a single transaction. In this paradigm, a user can instantiate and execute a contract in one step by sending a transaction that embeds both the contract code and execution parameters. This paradigm shift significantly reduces the resource consumption and time overhead of IoT devices during cross-chain interactions. Since the contract code does not need to permanently reside on the blockchain and is instantiated only at execution time, this approach naturally optimizes contract lifecycle management. Consequently, it is particularly suitable for IoT cross-chain payment scenarios where contracts are executed infrequently or on a one-time basis.

2.4 Zero-Knowledge Proofs

A zero-knowledge proof enables a prover to demonstrate that a statement is correct while disclosing no information beyond the fact of its validity. Its core properties include completeness, zero-knowledge, and soundness [15]. Among various ZKP constructions, zk-SNARKs stands out as an efficient non-interactive variant characterized by its small proof size and fast verification speed [16, 17]. A typical zk-SNARKs system consists of three algorithms: KeyGen, which generates the proving and verification keys; Proof, which produces the cryptographic proof and Verify, which validates the correctness of the proof [18]. In the proposed protocol, zk-SNARKs serves as an optional privacy-enhancing layer, providing strong confidentiality guarantees for sensitive information such as transaction amounts and participant identities.

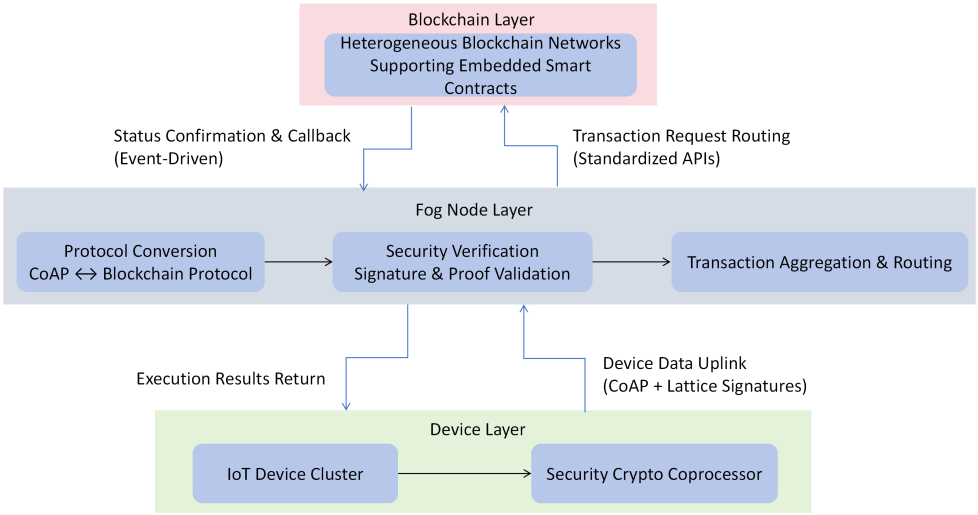

Fig. 1. System Model.

2.5 Constrained Application Protocol (CoAP)

As a lightweight IoT protocol, CoAP leverages UDP for transport and relies on DTLS to ensure secure data exchange. Compared with the Message Queuing Telemetry Transport (MQTT), CoAP supports request/response semantics and URI-based resource addressing [19, 20]. Its small header format and asynchronous messaging model make it suitable for constrained networks [21]. The proposed protocol employs CoAP as the communication interface connecting IoT devices and fog nodes, which facilitates efficient network performance when initiating cross-chain payments.

3 Model and design goal

3.1 System Model

This paper presents a lattice-based standardized cross-chain payment protocol that features a three-tier distributed architecture, as illustrated in Fig. 1, aiming to provide a comprehensive interoperability solution for the post-quantum IoT ecosystem. This architecture consists of the device layer, fog node layer, and blockchain layer from bottom to top. Each layer collaborates through standardized interfaces and protocols to collectively build a secure and efficient cross-chain payment environment.

The device layer consists of resource-constrained IoT devices that initiate and participate in cross-chain transactions. Each device is equipped with a lattice cryptography coprocessor for locally

executing Dilithium signatures and generating lightweight zk-SNARK proofs, ensuring authentication and privacy at the source.

The fog node layer connects devices to the blockchain network. It aggregates small transactions into batches for efficiency, verifies zk-SNARK proofs, and facilitates seamless conversion between the CoAP protocol and blockchain protocols. Fog nodes also provide caching for frequently accessed smart contract states and verification results, optimizing performance.

The blockchain layer comprises multiple heterogeneous blockchain networks, each supporting embedded smart contracts and standardized cross-chain message formats. It maintains the global state, ensuring immutability and traceability of transactions. The layer includes a smart contract engine, consensus mechanism, cross-chain communication module, and state management, enabling interoperability and automated payment logic.

3.2 Security Model

This protocol defines a formal security model to outline its security boundaries. It assumes that the Dilithium signature scheme and zero-knowledge proofs (e.g., zk-SNARKs) are secure under quantum computing, and that the participating blockchain networks are secure and active. The model considers a powerful adversary capable of passive eavesdropping, active message injection, tampering, replay, and collusion by malicious internal nodes, potentially with future quantum computing power. The adversary aims to steal privacy, disrupt service availability, commit fraud, or undermine the cryptographic foundation. In response, the protocol achieves three core security goals: post-quantum security, ensuring cryptographic operations are resistant to quantum attacks based on lattice-hardness assumptions; strong privacy protection, selectively hiding transaction identities and sensitive data using zero-knowledge proofs; and cross-chain atomicity, ensuring payments are either fully completed or rolled back on all involved blockchains, protecting user assets from cross-chain transaction failures.

3.3 Design Goal

The protocol aims to deliver a secure, efficient, and practical cross-chain payment solution for the post-quantum IoT ecosystem. Its design objectives are as follows:

Post-quantum security: All cryptographic operations, including authentication and transaction signing, are based on provably secure lattice primitives (e.g., the Dilithium signature scheme) to ensure resistance to quantum attacks. A smart contract–driven state coordination mechanism guarantees transaction atomicity, ensuring all cross-chain payments either succeed entirely or roll back completely, preventing asset loss. The protocol also integrates zero-knowledge proofs for optional privacy enhancement, protecting transaction identities and sensitive data.

Lightweight and efficient: By combining embedded smart contracts with streamlined communication mechanisms, the protocol minimizes both delay and energy usage, allowing it to satisfy the stringent timing and resource limitations of IoT systems.

Interoperable and scalable: Standardized interfaces and communication mechanisms enable secure value and data exchange across heterogeneous blockchains. The architecture supports dynamic

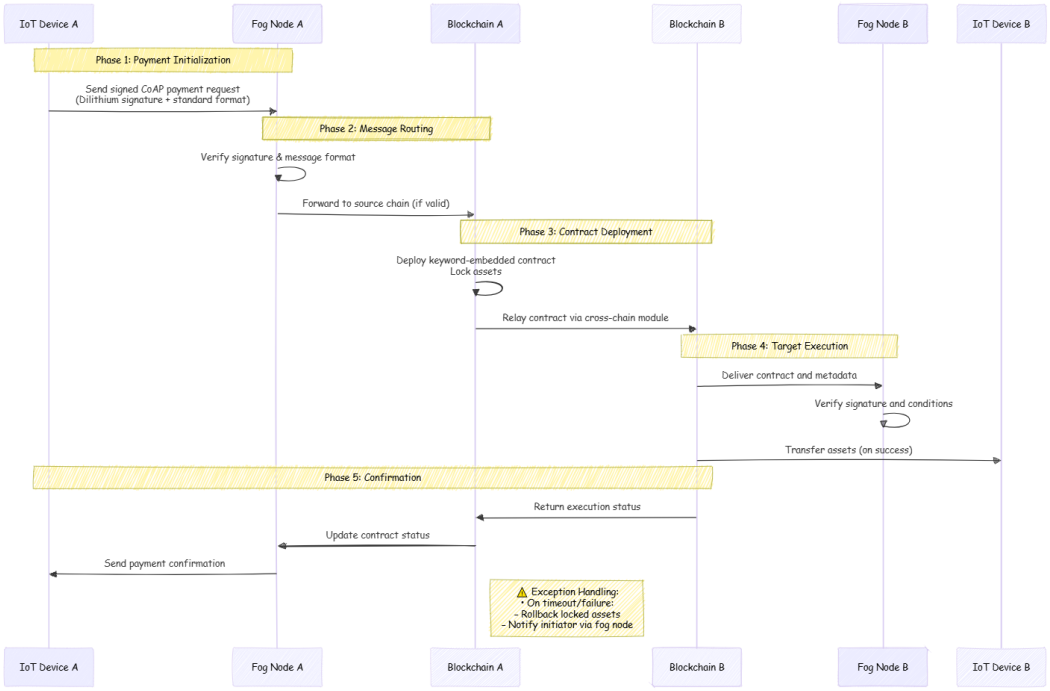

Fig. 2. Protocol Flowchart.

device access and high-frequency transactions, ensuring elastic scalability and forming a solid foundation for large-scale IoT integration.

4 Design of a Lattice-based Standardized Cross-chain Payment Protocol

4.1 Protocol Overview

The lattice-based standardized cross-chain payment protocol adopts a layered architecture to achieve atomic and secure cross-chain payments through a serialized interaction process. As shown in Fig. 2, the protocol comprises five core phases—payment initialization, cross-chain message routing, source-chain contract deployment, target-chain verification and execution, and result confirmation—supported by a comprehensive exception handling mechanism.

A core novelty of the design is its use of lattice-based security primitives together with a keyword-oriented embedded contract mechanism. Through standardized message formats and state machine coordination, the protocol ensures efficient and post-quantum-secure cross-chain value transfer. The design also considers the resource constraints of IoT devices, guaranteeing feasibility

in low-power environments.

4.2 Payment Initialization

In the payment initialization phase, the IoT device acts as the payment initiator, responsible for constructing a standardized payment request and performing local security verification. The core objective of this phase is to establish a complete payment credential on the device side, including generating a post-quantum secure digital signature and constructing a message format compliant with standardized specifications.

Algorithm 1 PaymentInitiation(λ, payment_params)

if key_pair = $\emptyset$ **then**
  $(pk, sk) \leftarrow \mathsf{Dilithium.KeyGen}(\lambda)$
end if
std_msg $\leftarrow$ ConstructStandardMessage(payment_params)
$\sigma \leftarrow \mathsf{Dilithium.Sign}(sk, \text{std\_msg})$
secure_pkg $\leftarrow$ {std_msg, σ, pk, device_id}
response $\leftarrow$ CoAP.Send('/payment/init', secure_pkg)
return (transaction_id, response)

Algorithm 1 initiates the protocol flow, constructing a secure payment request on resource-constrained IoT devices. It leverages lattice-based cryptography (Dilithium) to generate a digital signature at the source, ensuring integrity and non-repudiation, thus establishing a post-quantum secure foundation for the cross-chain process. By using a standardized message and the lightweight CoAP protocol, the algorithm effectively addresses the security, efficiency, and interoperability needs of IoT scenarios.

4.3 Cross-chain Message Routing

The fog node functions as an intelligent intermediary between IoT devices and the blockchain network, responsible for validating and routing payment requests. Upon receiving a secure package, it performs lattice-based signature verification using the Dilithium algorithm to ensure message authenticity and integrity, followed by standardized format validation to filter out malformed requests. Verified messages are then routed to the corresponding blockchain according to the target chain identifier. This layered process establishes a secure and modular gateway that isolates device management from blockchain complexity, enhancing both system security and operational efficiency.

4.4 Source-chain Contract Deployment and Keyword Management

Upon receiving the payment request, the source blockchain network deploys a keyword-embedded smart contract to manage the entire cross-chain payment lifecycle. This phase integrates lattice-based cryptographic security with keyword-driven state management to achieve secure and

Table 1: Keyword Definitions of the Keyword-Embedded Smart Contract

Keywords	Type	Default Value	Description
require	Numerical	0	Specifies the quantity of target assets needed for a cross-chain asset swap
provide	Numerical	0	Available assets from this contract for the cross-chain swap
completed	Boolean	false	Indicates whether the contract has been executed
paired	Boolean	false	Indicates whether a matching counterpart contract has been found
timeout	Numerical	0	Sets the contract timeout duration (unit: Δ)
disable	Boolean	false	Indicates whether the contract has been disabled
cross-chain	Boolean	false	Indicates whether the contract requires cross-chain interaction
chainID	String	null	Specifies the identifier of the target blockchain for interaction
targetAddress	String	null	Specifies the recipient address or contract address on the target chain
exchangeID	String	null	Contract identifier for the cross-chain asset exchange
runOnlyOnce	Boolean	false	Determines if the contract is designated for one-time execution only

efficient asset locking and state tracking. During contract instantiation, predefined keywords are initialized to specify state attributes and execution conditions. As listed in Table 1, these keywords collectively form a fine-grained state machine that governs the cross-chain payment process with precision.

A typical cross-chain payment contract progresses through the following sequence of states:

Initialization and Asset Locking: The contract is created with cross-chain set to true, and the require and provide parameters initialized according to the payment request. The corresponding assets are locked on the source chain. At this stage, completed and paired are false, while disable is also false.

Matching and Execution: The source-chain contract information is routed to the target chain. When a matching contract appears on the target chain (i.e., require and provide are complementary and exchangeID matches), the paired keyword for both contracts is set to true, enabling contract execution.

Completion Confirmation: Once the paired flag is enabled, the contract moves into its execution phase. After the target chain validates the request and finalizes the asset transfer, the source chain updates the completed field to true. Depending on the runOnlyOnce setting, the contract is then marked as disabled, and the locked assets are finally released to the designated recipient.

Exception/Timeout: If no successful matching or execution occurs within the timeout period Δ defined by timeout, the system automatically sets disable to true and triggers the asset rollback logic, returning the locked assets to the initiator. This mechanism ensures that assets are never lost under any failure condition and serves as the core guarantee of cross-chain atomicity.

Through the runOnlyOnce keyword, the contract can automatically enter a disabled state upon

completion, realizing autonomous lifecycle management. This design effectively reduces blockchain storage overhead, making it particularly suitable for IoT scenarios with frequent micro-payments.

Algorithm 2 represents the core innovation of the protocol, responsible for instantiating and managing the lifecycle of cross-chain payments on the source chain. By initializing a set of predefined keywords, it transforms the payment contract into a well-defined state machine. Its "lock-assets-first, then-trigger-cross-chain" execution logic serves as the key mechanism ensuring transactional atomicity. The algorithm's behavior is driven by operation outcomes that dynamically update keyword states, thereby achieving automated process control and self-adaptive fault management.

Algorithm 2 KeyContractDeployment(payment_data,σ)

keywords $\leftarrow$ InitializeKeywords(*payment_data*)
if $\neg$LockAssets(*payment_data.sender*, *payment_data.amount*) **then**
  return (error, 'ASSET_LOCKING_FAILED')
end if
cross_chain_msg $\leftarrow$ PrepareCrossChainMessage(*payment_data*, σ, keywords)
result $\leftarrow$ CrossChainSend(*payment_data.target_chain_id*, cross_chain_msg)
if *result.success* **then**
  UpdateKeyword('paired', True)
  return (success, *payment_data.transaction_id*)
else
  UnlockAssets(*payment_data.sender*, *payment_data.amount*)
  UpdateKeyword('disable', True)
  return (error, 'CROSS_CHAIN_INIT_FAILED')
end if

4.5 Target-Chain Verification and Execution

After receiving a cross-chain payment request, the target blockchain network performs comprehensive verification and asset transfer operations. This phase ensures the legitimacy and integrity of the request through layered cryptographic validation based on lattice signatures. Specifically, the target chain independently verifies the integrity hash of the received message, authenticates the Dilithium signature, and checks the validity of embedded payment conditions. Only when all verifications are successfully completed does the system execute the corresponding asset transfer to the designated recipient. This independent verification and execution process ensures that target-chain operations strictly comply with predefined rules, thereby preserving cross-chain atomicity and maintaining overall system consistency.

4.6 Result Confirmation and State Synchronization

After the cross-chain payment execution is completed, the system enforces final transaction consistency through a standardized confirmation mechanism. At this point, the fog node manages the synchronization of states across the source and target chains and updates the status of the keyword-

based smart contract accordingly. If the target chain confirms successful execution, the protocol finalizes asset settlement on the source chain and records the transaction as completed. Conversely, if an error occurs, the system performs a secure refund and marks the transaction as failed. This deterministic "success-or-refund" mechanism ensures that no assets remain locked or lost, thereby maintaining atomicity and reliability across chains.

4.7 Exception Handling and Recovery Mechanism

To ensure robustness and atomicity in distributed heterogeneous environments, the protocol incorporates an automated exception handling and recovery mechanism. By assigning a timeout period Δ to each cross-chain payment contract, the system automatically marks the contract as timed out and triggers asset rollback if a transaction fails to complete within the specified time—due to issues such as network interruptions or target-chain failures—thereby fundamentally preventing permanent asset locking. For transient exceptions (e.g., network congestion), the protocol employs a limited retry mechanism with an exponential backoff strategy to improve transaction success rates. Furthermore, through the keyword-based smart contract state machine and periodic inter-chain state verification, the system can detect and correct rare cases of state inconsistency, ensuring that all participating blockchains maintain final state consistency and providing strong fault tolerance for the entire cross-chain payment process.

4.8 Protocol Standardized Interface

To enable seamless integration and interoperability among heterogeneous IoT devices and diverse blockchain platforms, the protocol defines a lightweight, standardized RESTful API set. This interface specification clearly defines the request and response formats for core functions such as payment initialization, status query, and contract deployment. By encapsulating the underlying complexities of post-quantum cryptographic operations and cross-chain routing logic, the interface provides a simple and unified invocation method for upper-layer applications. Consequently, resource-constrained IoT devices can securely initiate and manage cross-chain payments via simple CoAP or HTTP requests, significantly reducing system integration complexity while ensuring scalability and usability across the entire ecosystem at the protocol level.

5 Security Analysis

A formal security assessment of the lattice-based standardized cross-chain payment protocol is conducted herein, considering a quantum adversary model. The attacker is assumed to be polynomially bounded and capable of intercepting or modifying messages, corrupting participants, and launching man-in-the-middle, replay, or collusion attacks. The security guarantees are founded upon three pillars: the quantum hardness of M-LWE/M-SIS, the EUF-CMA resilience of Dilithium signatures under the QROM and the collision resistance of the underlying hash functions.

5.1 Post-Quantum Security

Theorem 1. *The protocol ensures security against quantum adversaries through its lattice-based cryptographic component, which relies on the hardness of the M-LWE problem.*

Proof. The security reduction proceeds as follows. Suppose a quantum adversary $\mathscr{A}$ succeeds in attacking the protocol with advantage $\varepsilon(\lambda)$. Then one can construct an algorithm $\mathscr{B}$ that uses $\mathscr{A}$ to solve a given M-LWE instance. Given an M-LWE instance $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e}), (\mathbf{A} \in \mathbb{R}^{k\times\ell}$, where $\mathbf{s}$ and $\mathbf{e}$ are sampled from bounded distributions, $\mathscr{B}$ embeds $\mathbf{A}$ as the Dilithium public key matrix and incorporates a transformation of $\mathbf{b}$ into the public key. It then simulates the cross-chain payment environment with keyword-embedded contracts and standardized message routing. Upon receiving a signature query for a message m, $\mathscr{B}$ obtains a valid signature from the Dilithium signing oracle. If $\mathscr{A}$ forges a valid signature or derives the session key, $\mathscr{B}$ outputs the corresponding result, thereby solving M-LWE. Since M-LWE is hard in the quantum model and the protocol strictly adheres to the Dilithium parameters, it follows that

$$\Pr[\mathscr{B} \text{ solves M-LWE}] \geq \frac{\Pr[\mathscr{A} \text{ breaks the protocol}]}{\text{poly}(\lambda)} - \mathsf{negl}(\lambda). \tag{2}$$

Hence, $\Pr[\mathscr{A}$ breaks the protocol$] \leq \mathsf{negl}(\lambda)$. □

5.2 Standardized Cross-Chain Atomicity

Theorem 2. *Under the security of the lattice-based signature scheme and the standardized message format, the proposed protocol guarantees atomicity of payment operations across heterogeneous blockchains—that is, all related operations either commit together or abort together.*

Proof. The protocol state S is defined as the pair (S_A, S_B), where S_A and S_B correspond to the states on the source and target chains. In the normal execution, S_B switches from *pending* to *committed* only if

$$\mathsf{Dilithium.Verify}(pk, \{M_{\text{std}}\}, \sigma) = 1 \tag{3}$$

where M_{std} is the standardized cross-chain message. The confirmation returned to the source chain triggers S_A to update from locked tocommitted, completing asset release. If no valid confirmation is received within timeout period Δ, S_A automatically transitions to aborted according to the timeout keyword, initiating rollback. Therefore, the system cannot reach inconsistent states such as $S_A = \textit{committed}$ and $S_B = \textit{aborted}$. The probability of such inconsistency under a valid signature is negligible:

$$\Pr\left[\left(S_A^{\text{final}} \neq S_B^{\text{final}}\right) \wedge \text{ValidSignature}(tx)\right] \leq \mathsf{negl}(\lambda) \tag{4}$$

Hence, the atomicity property holds. □

5.3 Resistance to Key-Recovery Attacks

Theorem 3. *Under the hardness assumptions of the M-LWE and M-SIS problems, the proposed protocol prevents any adversary from recovering valid private keys from public information.*

Proof. For a key-recovery adversary $\mathscr{A}_{key}$, recovering $sk = (\mathbf{s}_1, \mathbf{s}_2)$ from $pk = (\mathbf{A}, \mathbf{t}_1)$ is infeasible under standard Dilithium parameters and the M-LWE assumption, even if partial blockchain compromise occurs:

$$\Pr[KeyRecovery] \leq \mathsf{negl}(\lambda) \tag{5}$$

For a forgery adversary $\mathscr{A}_{forge}$, each cross-chain message includes standardized fields (source-chain ID, target-chain ID), preventing replay on incorrect chains. Forging a valid signature would violate the EUF-CMA security of Dilithium:

$$\Pr[ForgeCrossChain] \leq \Pr[ForgeDilithium] + \mathsf{negl}(\lambda) \tag{6}$$

For error-based attacks, deterministic and randomized Dilithium signatures eliminate exploitable bias, and uniform verification across participants ensures consistency:

$$\Pr[ErrorAttack] \leq \mathsf{negl}(\lambda) \tag{7}$$

Therefore, the protocol achieves provable resistance against key-recovery attacks. □

5.4 Standardized Interface Security

Theorem 4 (Interface Security)**.** *Under lattice-based cryptographic assumptions, the standardized interface design of the protocol prevents security flaws arising from implementation inconsistencies, ensuring secure and interoperable cross-chain operations.*

Proof. Let $\mathscr{I}_{std} = \{\mathsf{PaymentInit}, \mathsf{CrossChainRoute}, \mathsf{VerifyPayment}\}$ denote the standardized interface set, where each implementation must follow a unified message format M_{std}. Each interface call is authenticated via a Dilithium signature, binding every request to specific chains and participants. A successful interface attack implies either a forged lattice-based signature or a violation of deterministic message parsing. The adversary's success probability satisfies:

$$\Pr[\text{InterfaceAttack}] \leq \Pr[\text{ForgeSignature}] + \Pr[\text{FormatExploit}] \tag{8}$$

Since the format parsing is deterministic and publicly verifiable, $\Pr[FormatExploit] \leq \mathsf{negl}(\lambda)$. Therefore,

$$\Pr[InterfaceAttack] \leq \mathsf{negl}(\lambda) \tag{9}$$

which concludes the proof. □

6 Performance Evaluation

The performance of the proposed lattice-based cross-chain protocol is evaluated in this section. Our simulation environment, hosted on a platform with an Intel Core i7-12700H CPU and 16 GB RAM, employs Docker containers to emulate the network of IoT devices, fog nodes, and blockchain nodes. For faster transaction processing, a Ganache test chain with minimal mining difficulty is utilized, enabling second-level block creation. Integrating the liboqs and libcoap libraries for cryptography and communication, the protocol's efficiency, end-to-end performance, and scalability are benchmarked against 2,000 simulated transactions.

Table 2: Cryptographic Operation Performance Benchmark (Unit: ms)

Metric	Dilithium2	Falcon-512	ECDSA
Key Generation	0.32 ± 0.02	36.12 ± 2.34	0.72 ± 0.05
Signature Generation	0.79 ± 0.06	6.28 ± 0.42	0.58 ± 0.04
Signature Verification	0.23 ± 0.02	0.09 ± 0.01	0.01 ± 0.001
Complete Operation	1.34 ± 0.08	42.49 ± 2.67	1.31 ± 0.07

6.1 Cryptographic Operation Benchmark

This experiment measures the execution efficiency of the protocol's core cryptographic primitives under typical IoT configurations. As summarized in Table 2, the post-quantum Dilithium2 algorithm achieves performance comparable to the classical ECDSA scheme while maintaining quantum resistance, completing a full signing and verification process in only 1.34 ms. In contrast, Falcon-512 shows noticeably higher computation time due to complex floating-point operations. These results, consistent with NIST PQC benchmarks, confirm that Dilithium2 offers a practical balance between efficiency and post-quantum security.

6.2 End-to-End Performance

The end-to-end evaluation compares the proposed protocol with representative cross-chain mechanisms, including hash time-lock contracts (HTLC), notary-based protocols, and a Falcon-based variant. As shown in Fig. 3, the proposed scheme achieves consistently lower transaction latency and higher throughput. The improvement mainly stems from keyword-embedded smart contracts and standardized message formats, which reduce redundant execution and communication overhead.

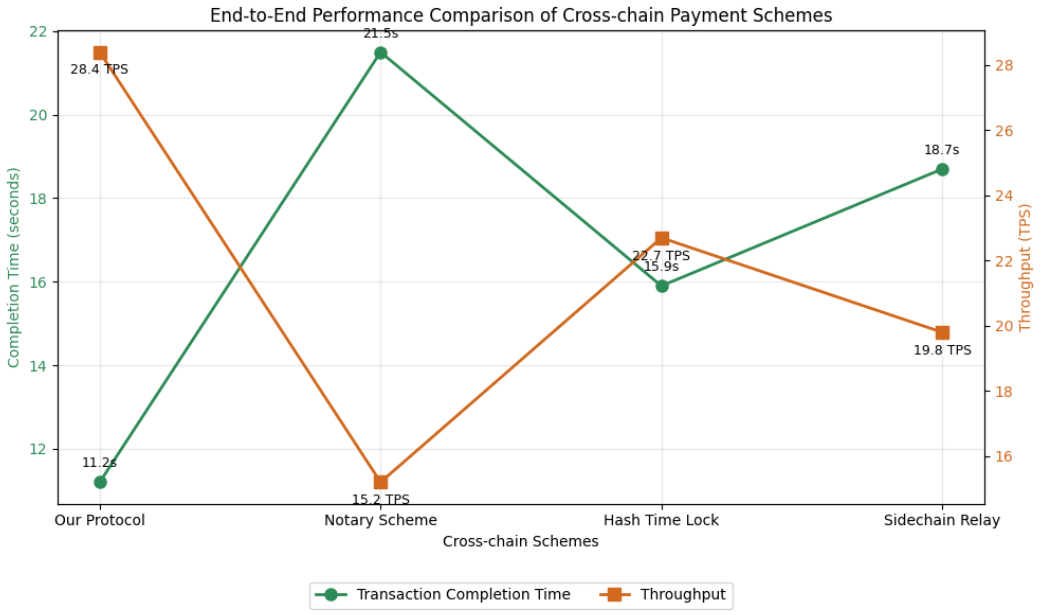

Fig. 3. End-to-End Performance Comparison.

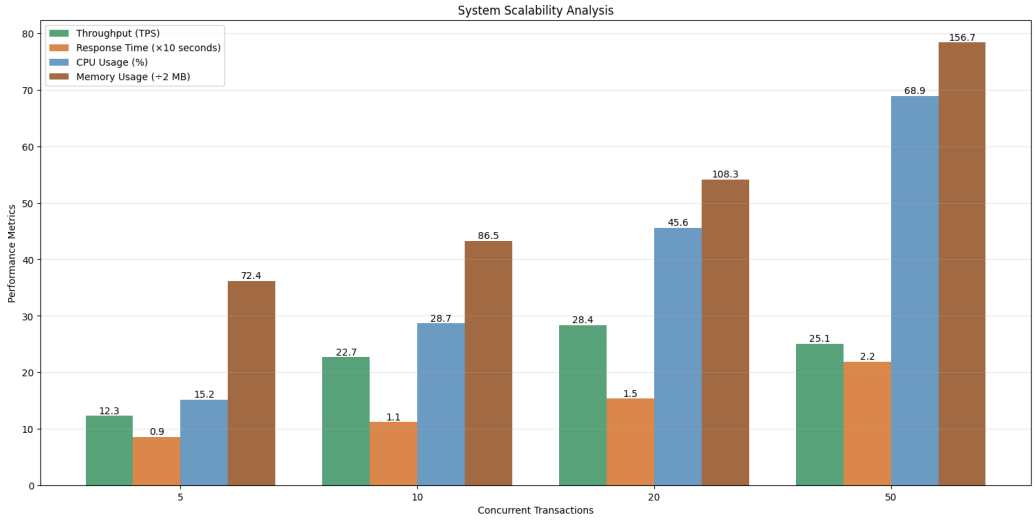

Fig. 4. System Scalability Analysis.

6.3 System Scalability

Scalability was tested by varying the number of concurrent transactions. As illustrated in Fig. 4, throughput increases nearly linearly at low-to-moderate loads and peaks at around 20 concurrent transactions, while CPU and memory usage remain stable. These results demonstrate that the fog-node aggregation and lightweight contract execution effectively maintain system efficiency under increasing workload.

Overall, the proposed protocol achieves a favorable trade-off among post-quantum security, computational efficiency, and scalability. Experimental results verify its suitability for deployment in large-scale, resource-constrained IoT environments.

7 Conclusion

With the continuous expansion of smart sensing, industrial automation, and machine-to-machine services, IoT systems are becoming increasingly dependent on secure and efficient value exchange across heterogeneous blockchain platforms. At the same time, the emergence of quantum computing poses a long-term threat to traditional public-key mechanisms, which makes the integration of post-quantum cryptography into IoT payment infrastructures an important research direction. In this context, this paper proposes a lattice-based standardized cross-chain payment protocol that integrates post-quantum cryptography with IoT architectures, addressing the dual challenges of multi-chain interoperability and quantum security. The protocol introduces a keyword-embedded smart contract mechanism based on lattice signatures and a standardized cross-chain state management model, whose security is formally proven under the quantum adversary model. Following a lightweight design principle, it optimizes cryptographic operations and communication protocols to enhance performance while maintaining post-quantum security. Experimental results show that the protocol achieves high transaction efficiency, low resource consumption, and good scalability, demonstrating its feasibility for real-world IoT deployment. Overall, this work provides a solid foundation for building secure, efficient, and scalable IoT value networks and offers a viable pathway for the post-quantum evolution of distributed systems.

Acknowledgments

This work was supported in part by the Colleges and Universities 20 Terms Foundation of Jinan City under Grant 202228093, in part by the Major Program of Shandong Provincial Natural Science Foundation for the Fundamental Research under Grant ZR2022ZD03, in part by the National Science Foundation of China under Grants 62272256 and 62202250, in part by the Shandong Province Youth Innovation Team Project under Grant 2024KJH032, in part by the National Natural Science Foundation of China under Grant 62402254, and in part by the Natural Science Foundation of Shandong Province of China under Grants ZR2022QF094 and ZR2022QF010.

References

[1] Dai HN, Zheng Z, Zhang Y. Blockchain for Internet of Things: A survey. IEEE internet of things journal. 2019;6(5):8076-94.

[2] Ferrag MA, Derdour M, Mukherjee M, Derhab A, Maglaras L, Janicke H. Blockchain technologies for the internet of things: Research issues and challenges. IEEE Internet of Things Journal. 2018;6(2):2188-204.

[3] Zhao Y, Chai Z, Li Y, Huang H, Kang H. Multi-Objective Computation Offloading based on Decentralized Deep Reinforcement Learning in Industrial Internet of Things. IEEE Transactions on Cognitive Communications and Networking. 2024.

[4] Zhang H, Su H, Wu X, Yang Y. Cross-chain interoperability and collaboration for keyword-based embedded smart contracts in internet of things. IEEE Internet of Things Journal. 2023;11(6):10791-807.

[5] Gaži P, Kiayias A, Zindros D. Proof-of-stake sidechains. In: 2019 IEEE Symposium on Security and Privacy (SP). IEEE; 2019. p. 139-56.

[6] Dai B, Jiang S, Zhu M, Lu M, Li D, Li C. Research and implementation of cross-chain transaction model based on improved hash-locking. In: International Conference on Blockchain and Trustworthy Systems. Springer; 2020. p. 218-30.

[7] Liu Y, Xu S, Yue Z. A Multi-Authority CP-ABE Scheme with Fully Outsourced Computation and Direct Attribute Revocation Based on the R-LWE Problem in Edge Computing. IEEE Internet of Things Journal. 2025.

[8] Zhou Z, He D, Liu Z, Luo M, Choo KKR. A software/hardware co-design of crystals-dilithium signature scheme. ACM Transactions on Reconfigurable Technology and Systems (TRETS). 2021;14(2):1-21.

[9] Ducas L, Kiltz E, Lepoint T, Lyubashevsky V, Schwabe P, Seiler G, et al. Crystals-dilithium: A lattice-based digital signature scheme. IACR Transactions on Cryptographic Hardware and Embedded Systems. 2018:238-68.

[10] Prest T. Sharper bounds in lattice-based cryptography using the Rényi divergence. In: International Conference on the Theory and Application of Cryptology and Information Security. Springer; 2017. p. 347-74.

[11] Peikert C. Lattice cryptography for the internet. In: International workshop on post-quantum cryptography. Springer; 2014. p. 197-219.

[12] Brandstätter T, Schulte S, Cito J, Borkowski M. Characterizing efficiency optimizations in solidity smart contracts. In: 2020 IEEE International Conference on Blockchain (Blockchain). IEEE; 2020. p. 281-90.

[13] Su H, Guo B, Shen Y, Zhang Z, Qin C. To delay instantiation of a smart contract to save calculation resources in IoT. Wireless Communications and Mobile Computing. 2021;2021(1):6666236.

[14] Su H, Guo B, Shen Y, Suo X. Embedding smart contract in blockchain transactions to improve flexibility for the IoT. IEEE Internet of Things Journal. 2022;9(19):19073-85.

[15] Goldwasser S, Micali S, Rackoff C. The knowledge complexity of interactive proof-systems. In: Providing sound foundations for cryptography: On the work of shafi goldwasser and silvio micali; 2019. p. 203-25.

[16] Pinto AM. An introduction to the use of zk-SNARKs in blockchains. In: Mathematical Research for Blockchain Economy: 1st International Conference MARBLE 2019, Santorini, Greece. Springer; 2020. p. 233-49.

[17] Groth J. On the size of pairing-based non-interactive arguments. In: Annual international conference on the theory and applications of cryptographic techniques. Springer; 2016. p. 305-26.

[18] Ishai Y, Su H, Wu DJ. Shorter and faster post-quantum designated-verifier zkSNARKs from lattices. In: Proceedings of the 2021 ACM SIGSAC conference on computer and communications security; 2021. p. 212-34.

[19] Thangavel D, Ma X, Valera A, Tan HX, Tan CKY. Performance evaluation of MQTT and CoAP via a common middleware. In: 2014 IEEE ninth international conference on intelligent sensors, sensor networks and information processing (ISSNIP). IEEE; 2014. p. 1-6.

[20] Ludovici A, Moreno P, Calveras A. TinyCoAP: A novel constrained application protocol (CoAP) implementation for embedding RESTful web services in wireless sensor networks based on TinyOS. Journal of Sensor and Actuator Networks. 2013;2(2):288-315.

[21] Sezer BB, Akleylek S. PPLBB: a novel privacy-preserving lattice-based blockchain platform in IoMT. Journal of Supercomputing. 2025;81(1).