Online-Journals.org

iJIM International Journal of Interactive Mobile Technologies

iJIM | eISSN: 1865-7923 | Vol. 20 No. 6 (2026) | OPEN ACCESS

https://doi.org/10.3991/ijim.v20i06.60367

PAPER

CIPHER-IoT: Zero-Knowledge Proof Integration with Hyperledger Fabric for Privacy-Preserving IoT Systems

Shahnawaz Qadir (✉), Rana Hashmy

University of Kashmir, Srinagar, India

sqadir@uok.edu.in

ABSTRACT

The proliferation of Internet of Things (IoT) devices creates unprecedented security, privacy, and transparency challenges in distributed systems. Traditional encryption-based approaches provide privacy but impose significant computational overhead, storage bloat, and key management complexity. This paper presents CIPHER-IoT, a blockchain-based framework that integrates Zero-Knowledge Proofs (ZKPs) with Hyperledger Fabric for privacy-preserving IoT data management. Unlike encryption-based approaches that store encrypted data on-chain, CIPHER-IoT utilises Groth16 zk-SNARKs to generate cryptographic proofs of data validity while storing only commitments on the blockchain, achieving stronger privacy guarantees with lower storage overhead. The framework employs Ed25519 for lightweight digital signatures and implements comprehensive chaincode for ZKP verification, commitment uniqueness checking, and access control enforcement. CIPHER-IoT targets gateway/edge IoT deployments with moderate computational capacity (ARM processors 500 MHz+) rather than ultra-constrained sensors. We evaluate CIPHER-IoT against two baseline systems, SPAS (homomorphic encryption-based) and SPAS-H (AES encryption with Hyperledger Fabric), using realistic simulation with 50–500 devices and transaction rates of 10–75 TPS. Experimental results demonstrate superior privacy (98% confidentiality vs. 80–95% for encryption-based approaches) alongside competitive performance: read latency improves 37% ($p < 0.001$), throughput increases 14.6% ($p < 0.001$), memory reduces 21.4%, network bandwidth saves 47%, and disk I/O reduces 37.8%. The system maintains zero data loss under failure scenarios and scales linearly to 500 devices with minimal degradation (9.9%). CIPHER-IoT demonstrates that verification-based privacy mechanisms can achieve stronger privacy and better performance than transformation-based approaches in distributed validation contexts, particularly suitable for enterprise IoT deployments requiring coordinated privacy-preserving infrastructure.

KEYWORDS

Internet of Things (IoT), hyperledger fabric, blockchain, zero-knowledge proofs, groth16 zk-SNARKs, privacy-preserving systems, cryptographic commitments, access control

Qadir, S., Hashmy, R. (2026). CIPHER-IoT: Zero-Knowledge Proof Integration with Hyperledger Fabric for Privacy-Preserving IoT Systems. *International Journal of Interactive Mobile Technologies (iJIM)*, 20(6), pp. 89–112. https://doi.org/10.3991/ijim.v20i06.60367

Article submitted 2025-12-30. Revision uploaded 2026-02-16. Final acceptance 2026-02-17.

1 INTRODUCTION

The Internet of Things (IoT) has fundamentally transformed how devices communicate and generate value across interconnected networks. With over 75 billion connected IoT devices now deployed globally, the ecosystem continues to expand across diverse domains, including smart cities, healthcare, industrial automation, and consumer electronics [1]. This exponential growth has created unprecedented opportunities for data-driven decision-making and intelligent services. However, the same interconnectivity that powers these benefits introduces critical challenges in security, privacy, and transparency that threaten the viability of IoT ecosystems. IoT deployments span a spectrum from ultra-constrained sensors (8-bit microcontroller, < 100 MHz, milliwatt power budget) to gateway/edge devices (32–64-bit processor, 500 MHz-2 GHz, multi-watt operation). CIPHER-IoT (Cryptography-Integrated Platform for Hyperledger-Enabled Robustness in IoT Systems) targets the latter category, which represents the increasingly common deployment model in modern enterprise IoT systems where computational resources enable cryptographic operations.

Modern IoT systems face a fundamental trilemma, where achieving security, privacy, and transparency simultaneously appears mutually exclusive. Security vulnerabilities arise from resource-constrained devices with limited computational capabilities deployed in physically accessible locations, making traditional security mechanisms designed for resource-rich environments often infeasible [1]. Privacy concerns emerge from the continuous generation of sensitive data ranging from health vitals in wearable devices to behavioural patterns in smart homes, creating tension between data confidentiality and analytical utility [2]. Transparency requirements for critical applications demand verifiable audit trails and accountability, yet transparency mechanisms typically expose sensitive information, directly conflicting with privacy objectives [3].

Traditional approaches to IoT security rely on established architectural patterns, each with inherent limitations. The centralised access control systems using Role Based Access Control (RBAC) create single points of failure, lack transparency in decision-making, and struggle to scale across organisational boundaries [1]. Encryption-based privacy mechanisms provide confidentiality but introduce significant overhead. Homomorphic encryption enables computation on encrypted data but incurs 100–1000 times the computational cost [4], while symmetric encryption requires complex key management infrastructure despite its efficiency. Public blockchains offer decentralisation and transparency but prove unsuitable for IoT due to energy-intensive probabilistic consensus, limited throughput of 7–15 transactions per second, high latency measured in minutes to hours, and complete data transparency without privacy [5].

Recent research has explored permissioned blockchains, particularly Hyperledger Fabric, combined with encryption for IoT applications [6], [7], [8], [9]. While these approaches achieve better performance than public blockchains and maintain some privacy, they still store encrypted data on-chain, resulting in storage bloat, slow query performance requiring decryption, and key management complexity. More fundamentally, these solutions typically optimise for two dimensions of the security-privacy-transparency trilemma while sacrificing the third, representing architectural trade-offs rather than comprehensive solutions.

Zero-Knowledge Proofs (ZKPs) offer a cryptographic breakthrough that potentially resolves this trilemma. ZKPs enable a prover to convince a verifier that a statement is true without revealing any information beyond the statement's validity. In

IoT contexts, devices can prove possession of valid sensor data satisfying specific conditions without revealing actual sensor readings or device credentials. Modern ZKP schemes, particularly zk-SNARKs (Zero-Knowledge Succinct Non-Interactive Arguments of Knowledge), provide properties ideal for IoT-blockchain integration: constant-size proofs regardless of statement complexity, non-interactive protocols eliminating round-trip communication overhead, millisecond-scale verification enabling real-time applications, and perfect privacy through zero-knowledge properties.

Despite these advantages, integrating ZKPs with permissioned blockchains for IoT remains largely unexplored. Existing ZKP-blockchain research focuses primarily on public blockchains such as Zcash and Ethereum with zk-rollups or presents theoretical frameworks without practical implementation and rigorous evaluation [10], [11]. Our comprehensive analysis of recent literature reveals several critical gaps. First, no practical ZKP-Hyperledger Fabric integration with working chaincode and realistic IoT simulation exists. Second, comparative analyses against established baselines with quantified privacy-performance trade-offs are absent. Third, implementation details and deployment methodologies are insufficiently documented. Fourth, privacy claims lack quantitative measurement frameworks for objective comparison. Fifth, scalability evaluations rarely extend beyond small-scale simulations or assess system behaviour under stress, failure, or attack conditions.

This paper addresses these gaps through CIPHER-IoT (Cryptography-Integrated Platform for Hyperledger-Enabled Robustness in IoT Systems), a novel comprehensive framework integrating Groth16 zk-SNARKs with Hyperledger Fabric for privacy-preserving IoT data management. The framework makes several key contributions. We present the first complete integration of Groth16 zk-SNARKs with Hyperledger Fabric, including custom chaincode implementing ZKP verification, commitment storage, and replay prevention, along with mathematical formalisation of the ZKP scheme adapted for IoT data verification. We detail the implementation approach, including a Python-based IoT simulator generating realistic sensor data, a Go-based chaincode architecture, a containerised deployment methodology, and the integration of Ed25519 for lightweight digital signatures. We conduct comprehensive performance evaluation with rigorous comparison against two baseline systems employing homomorphic and symmetric encryption, statistical analysis including 95% confidence intervals and hypothesis testing, five independent trial runs totalling many hours of continuous testing, and scalability evaluation from 50 to 500 simulated IoT devices. We introduce novel privacy metrics, including Verification Efficiency, Proof Complexity, and Confidentiality Level, enabling quantitative privacy assessment and comparative analysis. We demonstrate through privacy-performance analysis that ZKP-based approaches achieve both stronger privacy and better performance, challenging conventional wisdom about inherent privacy-performance trade-offs.

Our experimental evaluation demonstrates that CIPHER-IoT reduces read latency by 37% ($p < 0.001$), increases throughput by 14.6% ($p < 0.001$), and achieves 21.4% memory savings compared to encryption-based baseline systems, while simultaneously providing a 98% confidentiality level compared to 80–95% for conventional approaches. Resource utilisation analysis reveals a 47% network bandwidth reduction and 37.8% disk I/O reduction. The system maintains zero data loss under failure scenarios and scales linearly to 500 devices with minimal performance degradation. These results confirm that CIPHER-IoT achieves the security-privacy-transparency trilemma through cryptographic innovation rather than architectural compromises. The remainder of this paper is organised as follows: Section 2 reviews related work

and identifies research gaps; Section 3 presents the CIPHER-IoT architecture and implementation; Section 4 evaluates performance through comprehensive experiments; Section 5 concludes with implications and future directions.

2 RELATED WORK

Research on blockchain-based IoT security has evolved through several distinct approaches, each addressing different aspects of the security-privacy-transparency challenge. This section reviews relevant literature across blockchain-IoT integration, privacy-preserving mechanisms, and Hyperledger Fabric applications, identifying gaps that motivate CIPHER-IoT.

2.1 Blockchain for IoT data integrity and transparency

Early blockchain-IoT research focused primarily on leveraging distributed ledgers for data integrity and auditability. Jeong et al. [6] proposed a blockchain-based video surveillance framework leveraging Hyperledger Fabric's private data collections for metadata storage and encryption-based privacy. However, encryption-based approaches result in storage overhead and key management complexity. While this approach enhances confidentiality and auditability, it remains partially centralised due to reliance on internal managers, retaining the risk of insider compromise. Additionally, the encryption-based privacy approach results in storage overhead and key management complexity.

Yadav et al. [7] utilised blockchain technology and smart contracts to design a secure and transparent home rental system, aiming to eliminate third-party intermediaries and automate payment settlements. The framework operates using both on-chain and off-chain Hyperledger Fabric transactions, with off-chain processes managing agreements between parties and on-chain transactions handling payments and digital key generation. While the architecture enhances immutability and operational efficiency, it lacks robust privacy-preserving mechanisms, as anonymity and advanced cryptographic protections were not incorporated. This limitation restricts applicability in scenarios requiring strong privacy guarantees.

Blockchain has been extensively applied to supply chain management for transparency and traceability. Research by Zhong et al. [12] and others Bamakan et al. [9] demonstrates blockchain's effectiveness in establishing trust among suppliers, minimising intermediary dependence, and enabling real-time verification of product quality during transportation. These systems improve traceability and reduce costs but often involve high deployment expenses, manual verification overhead, and significant documentation requirements. More critically for privacy-sensitive applications, the transparency that makes blockchain valuable for supply chains becomes a liability when dealing with confidential business information or personal data.

IoT systems face increasing cyber threats requiring real-time detection mechanisms. Al-Haboosi et al. [13] proposed a hybrid transformer-based cyber-attack detection model for IoT networks using CNN and XGBoost, achieving 99.49% accuracy in identifying IoT hazards. This demonstrates that machine learning approaches can effectively detect both known and sophisticated attacks, addressing a critical security gap where traditional rule-based methods struggle with emerging threat patterns in resource-constrained IoT environments.

2.2 Privacy-preserving blockchain architectures

Recognising privacy limitations in transparent blockchain systems, researchers have explored various privacy-enhancing techniques. Shammar et al. [1] proposed an attribute-based access control model for IoT using Hyperledger Fabric, addressing scalability and transparency challenges in distributed IoT environments. Their work demonstrates the feasibility of fine-grained access control in permissioned blockchains but does not address privacy preservation of the data itself once access is granted. Similarly, Iftekhar et al. [2] developed an access control system for IoT layers in blockchain applications, focusing on authentication and authorisation mechanisms rather than data confidentiality.

Privacy-preserving techniques have been integrated with blockchain through various cryptographic mechanisms. Kurniawan et al. [8] developed a blockchain-based system for securing mobility data in smart campus environments, employing SHA256 hashing and proof-of-work mechanisms to protect personal location data through decentralisation, achieving 16.64 milliseconds block mining time with practical security validation. However, homomorphic encryption has been explored for privacy-preserving computation on blockchain data, enabling operations on encrypted values without decryption. While theoretically powerful, homomorphic encryption introduces prohibitive computational overhead for resource-constrained IoT devices, with operations requiring 100–1000 times more computation than plaintext operations [4]. This performance penalty makes homomorphic approaches impractical for real-time IoT applications requiring low latency.

2.3 Hyperledger fabric for enterprise IoT

Hyperledger Fabric has emerged as a leading permissioned blockchain framework for enterprise IoT applications due to its modular architecture, pluggable consensus mechanisms, and privacy features [5]. The framework's Membership Service Provider enables identity management through X.509 certificates, while Private Data Collections allow selective data sharing among authorised participants [14]. These features make Fabric particularly suitable for enterprise contexts requiring controlled access and regulatory compliance.

Recent work has leveraged Fabric's capabilities for various IoT domains. Abang et al. [15] examined latency performance modelling in the Hyperledger Fabric blockchain with an IoT perspective, identifying performance bottlenecks and optimisation opportunities. Ucbas et al. [4] conducted performance and scalability analysis comparing Ethereum and Hyperledger Fabric, demonstrating Fabric's superior throughput and lower latency for permissioned scenarios. Lee et al. [11] analysed latency characteristics of Hyperledger Fabric for blockchain-enabled IoT, providing modelling frameworks for predicting system behaviour under various loads.

Several researchers have integrated Fabric with IoT for specific applications. Hosseini et al. [5] reviewed blockchain-based decentralised identification in IoT, examining existing frameworks and their limitations in identity management and privacy preservation. El Ghazouani et al. [16] proposed an optimal approach combining blockchain technology with multi-agent systems to ensure data integrity and deduplication in cloud environments, demonstrating practical feasibility for managing large-scale data storage with batch auditing and deduplication support. Balasubramanian and Akila [17] implemented blockchain for the agricultural food supply chain using Hyperledger Fabric, addressing traceability and quality

assurance challenges. Li et al. [18] proposed a blockchain-based product traceability system with off-chain EPCIS (Electronic Product Code Information Services) and IoT device authentication, achieving scalability through hybrid on-chain and off-chain storage.

While these Fabric-based systems demonstrate the platform's versatility, they predominantly rely on encryption for privacy, storing encrypted data on-chain or in Private Data Collections [19]. This approach introduces storage overhead, requires complex key management, and necessitates decryption for data validation, creating performance bottlenecks [19]. Furthermore, existing work lacks comprehensive privacy metrics enabling quantitative comparison of different privacy-preserving approaches.

2.4 ZKPs in blockchain systems

Zero-Knowledge Proofs have gained prominence in blockchain privacy research, primarily in public blockchain contexts. Recent surveys [20] document the proliferation of ZKP applications across blockchain platforms, demonstrating the maturation of cryptographic techniques suitable for production deployments. Zcash pioneered the use of zk-SNARKs for private cryptocurrency transactions, demonstrating that users can prove transaction validity without revealing sender, receiver, or amount [10]. Ethereum's layer-2 scaling solutions increasingly employ ZKP-based rollups, batching thousands of transactions into single on-chain proofs to improve throughput while maintaining security guarantees.

However, ZKP integration in permissioned blockchains for IoT remains largely unexplored. Public blockchain ZKP implementations face different constraints than IoT applications: public blockchains prioritise censorship resistance over performance, accept high proof generation times measured in minutes, and assume powerful client hardware. In contrast, IoT requires low latency measured in milliseconds, must operate on resource-constrained devices, and benefits from permissioned trust models that simplify ZKP setup procedures.

Recent theoretical work has proposed ZKP-based privacy for IoT but lacks practical implementation. Proposed architectures typically omit critical details, including circuit design for IoT data validation, chaincode implementation for proof verification, performance evaluation with realistic IoT workloads, and comparative analysis against alternative privacy mechanisms. This gap between theoretical proposals and practical deployments motivates the need for comprehensive ZKP-Fabric-IoT integration with rigorous evaluation.

2.5 Research gaps and positioning of CIPHER-IoT

Analysis of existing literature reveals several critical gaps that CIPHER-IoT addresses [21]. First, no prior work demonstrates complete integration of ZKPs with Hyperledger Fabric for IoT, including working chaincode, a realistic simulation environment, and a comprehensive performance evaluation. Second, existing privacy-preserving blockchain-IoT systems lack quantitative privacy metrics enabling objective comparison of different approaches. Third, comparative evaluations against well-defined baseline systems with statistical rigour are absent, making it difficult to assess the relative merits of proposed solutions. Fourth, reproducibility is hindered by missing implementation details and incomplete experimental

procedures. Fifth, scalability evaluations rarely extend beyond small-scale simulations or assess system behaviour under failure conditions. Sixth, applicability constraints require explicit scoping; prior work rarely acknowledges technical maturity constraints or appropriate deployment contexts for proposed solutions.

CIPHER-IoT distinguishes itself through several unique contributions compared to related work. Unlike encryption-based approaches that store encrypted data on-chain [6], [7], [16], [17], [18], CIPHER-IoT stores only cryptographic commitments (32 bytes), while proofs verify data validity without revealing content, achieving stronger privacy with lower storage overhead. Commitment-based storage approaches [22], [23] have demonstrated practical effectiveness for IoT applications, providing compact data representation while maintaining cryptographic verification properties necessary for distributed consensus. Compared to public blockchain ZKP implementations [10], CIPHER-IoT leverages permissioned blockchain properties for simplified setup procedures, deterministic consensus eliminating verification uncertainty, and performance optimisation for IoT constraints. Unlike theoretical ZKP proposals [11], CIPHER-IoT provides complete implementation, including custom chaincode for proof verification, a Python-based IoT simulator with realistic workloads, a Docker-based reproducible testbed, and rigorous experimental evaluation with statistical analysis. Most significantly, CIPHER-IoT introduces quantitative privacy metrics and demonstrates through empirical evaluation that ZKP-based approaches achieve both stronger privacy (98% confidentiality vs. 80–95% for encryption) and better performance (37% lower read latency, 14.6% higher throughput), challenging conventional assumptions about privacy-performance trade-offs.

The comprehensive integration of ZKPs with Hyperledger Fabric for IoT, combined with rigorous experimental evaluation demonstrating both privacy and performance improvements, positions CIPHER-IoT as a significant advancement over existing blockchain-IoT security frameworks.

3 METHODOLOGY AND IMPLEMENTATION

3.1 System architecture

CIPHER-IoT integrates the Hyperledger Fabric blockchain with IoT devices through a layered architecture addressing centralisation, weak authentication, and single points of failure in conventional IoT systems. The architecture comprises four layers: IoT Device Layer with sensors and actuators, the Gateway Layer bridging devices to blockchain, the Blockchain Layer containing Hyperledger Fabric components, and the Storage Layer maintaining the immutable ledger and world state.

The Hyperledger Fabric network organises nodes into organisations managed by Membership Service Providers (MSPs) responsible for identity management via X.509 certificates. Peers maintain distributed ledgers and execute chaincode, with roles divided between endorsing peers validating transactions and committing peers updating ledger state. The ordering service employs Raft consensus, ensuring deterministic transaction sequencing with a 500 ms tick interval, a 5-second election timeout, and a 2-second batch timeout. The Raft consensus mechanism provides deterministic ordering with explicit fault tolerance semantics. The mechanism tolerates up to $\lfloor (n-1)/2 \rfloor$ node failures (2 failures in the 5-node orderer cluster) while maintaining consistency. During network partitioning, Raft's election mechanism enables recovery: minority partition nodes revert to the follower state, then resync with the majority partition to maintain transaction consistency. Experimental failure

recovery results (Section 4.3) validate this approach. Single peer failures recover in 2.3 seconds with 99.9% availability, while 30-second network partitions heal in 31.5 seconds with 99.1% availability. This predictable behaviour distinguishes CIPHER-IoT from public blockchain systems, where controlled network policies and node configurations in permissioned settings eliminate the need for probabilistic safety arguments. Channels enforce privacy by restricting visibility to authorised participants. Figure 1 illustrates the complete CIPHER-IoT architecture, showing communication flows between components.

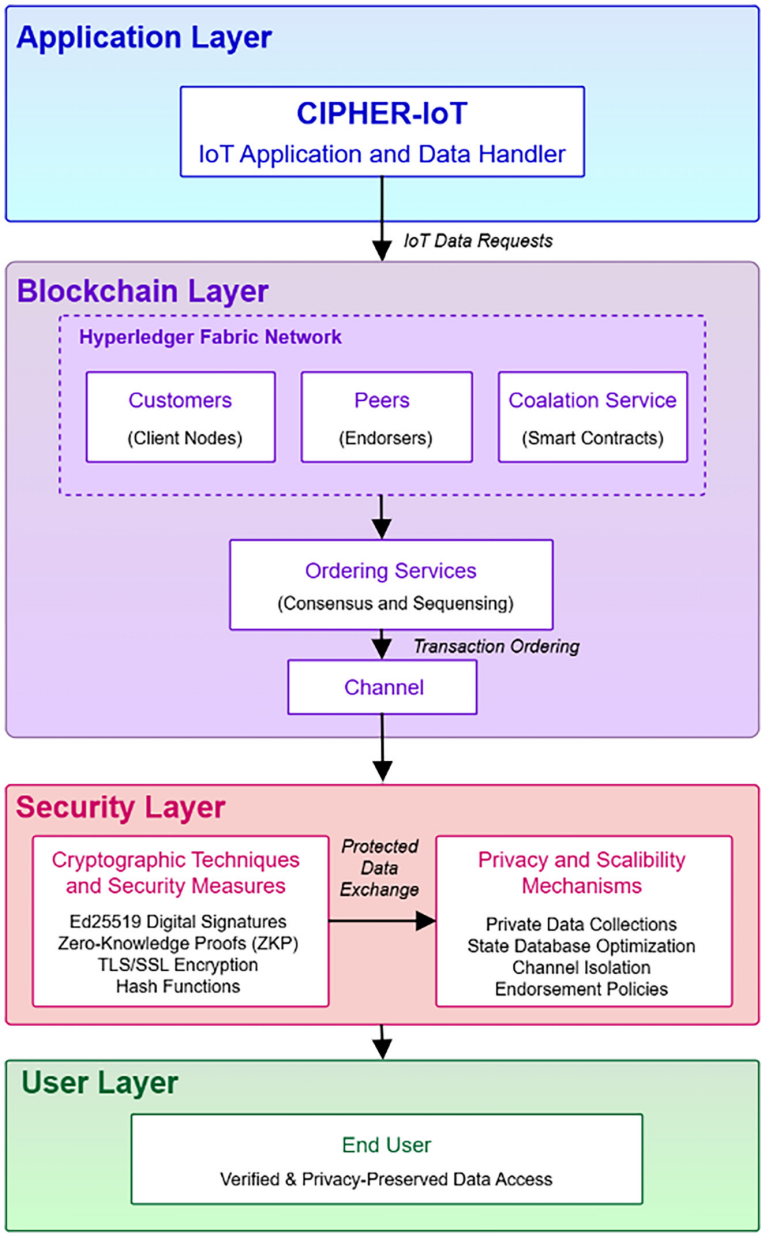

Fig. 1. Layered architecture of the CIPHER-IoT

The architecture integrates IoT devices with the Hyperledger Fabric blockchain through multiple layers: the IoT Device Layer (sensors and actuators), the Gateway Layer (data aggregation and protocol translation), the Blockchain Layer

(Hyperledger Fabric components including peers, orderers, and CAs), and the Storage Layer (world state and immutable ledger). TLS encryption secures all inter-layer communications.

3.2 Cryptographic framework

Lightweight Cryptography and security goals. Resource-constrained IoT devices require efficient cryptographic solutions balancing security and performance. CIPHER-IoT employs Ed25519 signatures, SHA-256 hashing, and commitment-based storage optimised for IoT constraints. Ed25519 provides equivalent security to 2048-bit RSA with 256-bit keys (8× reduction), constant-time verification resistant to side-channel attacks (~0.5 ms per signature on ARM Cortex-M4), and 64-byte signatures minimising transmission overhead. SHA-256 generates cryptographic commitments binding sensor data to blockchain transactions with 0.8 ms processing per kilobyte, constituting less than 0.3% of transaction latency. Security analysis confirms SHA-256 collision resistance via birthday bound (probability 2^{-128}) and preimage resistance requiring simultaneous collision of sensor data, device secrets, and hash values, computationally infeasible for resource-constrained adversaries. Nonce randomisation in witness construction ensures unique commitments, preventing replay attacks. These mechanisms provide data integrity, authentication, non-repudiation, privacy via commitment-based storage, and smart contract authorisation.

Zero-knowledge proof integration. We integrate Groth16 zk-SNARKs, providing constant 192-byte proofs with 4 ms verification time. The scheme comprises $Setup(C,\lambda) \rightarrow (pk,vk)$ for proving and verification key generation, $prove(pk,x,w) \rightarrow \pi$ for proof creation from public input x and private witness w, and $Verify(vk,x,\pi) \rightarrow \{0,1\}$ for validation. Security properties include completeness (valid proofs always verify), soundness (invalid statements cannot produce accepting proofs), and zero-knowledge (proofs reveal nothing beyond statement validity). In CIPHER-IoT's permissioned context, trusted setup governance is simplified through multi-party computation across organisation representatives, mitigating "toxic waste" risks. For deployments requiring transparent setup, zk-STARKs (Zero-Knowledge Scalable Transparent Argument of Knowledge) offer alternatives with trade-offs: 8 KB proofs (42× larger) and ~50 ms verification (12.5× slower) [22]. The current implementation prioritises performance for IoT edge nodes through Groth16's efficiency advantages.

The implementation employs simulated proof structures compatible with the Groth16 format for performance evaluation, with full methodological disclosure provided in Section 4.1. The threat model assumes adversaries can observe blockchain state, network traffic, and transactions but cannot: (1) compromise cryptographic assumptions (computational Diffie-Hellman for Groth16, preimage resistance for SHA-256); (2) access private witness data or CA key material; (3) compromise sufficient consortium key material. Security holds against passive adversaries with known-plaintext capability; active adversaries attempting replay attacks are detected through nonce randomisation and commitment uniqueness checking. Sybil attacks are mitigated through permissioned architecture with MSP-verified device identities.

The transaction flow extends standard Hyperledger Fabric with ZKP verification at endorsement. IoT simulators generate sensor data, compute SHA-256 commitments $h = Hash(w)$, generate ZKP proofs π, and submit transactions

$\tau = \{h, \pi, metadata, signature\}$. Endorsing peers verify signatures, validate ZKP format and binding, check commitment uniqueness to prevent replay attacks, validate access policies, and return endorsements for valid transactions. Ordering service sequences endorsed transactions into blocks. Committing peers re-verify endorsements, update the world state storing commitments and proofs, and append blocks to the immutable ledger.

Figure 2 illustrates the network architecture comprising two organisations (Org1 and Org2), each with endorsing and committing peers.

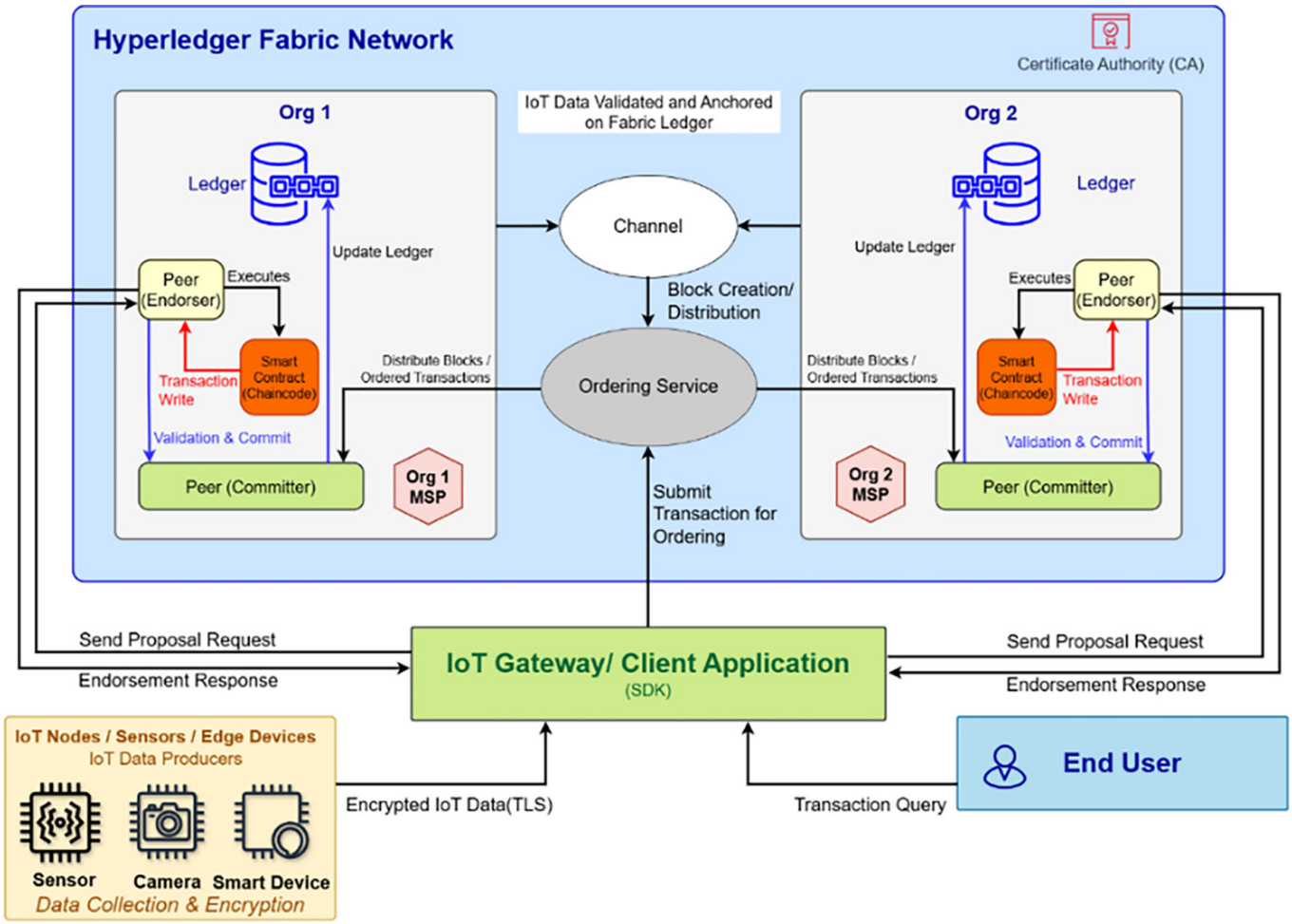

Fig. 2. Hyperledger Fabric network architecture in CIPHER-IoT

The network comprises two organisations (Org1, Org2), each with endorsing and committing peers. The ordering service coordinates transaction sequencing using Raft consensus. Smart contracts (chaincode) are deployed on endorsing peers, while end users interact via client applications. All communications are secured with TLS 1.3 mutual authentication.

Chaincode implements verification through seven sequential checks: 1) Ed25519 signature validation, 2) proof format verification, 3) proof-commitment binding, 4) commitment uniqueness via state database query (O (log n) lookup, 1–2 ms), 5) access policy validation, 6) timestamp freshness within a 5-minute window (prevents clock-skew replay attacks through NTP synchronisation) and 7) commitment storage with event emission. Multi-layer replay prevention combines nonce-based

randomisation ensuring unique commitments for identical sensor readings, commitment uniqueness enforcement preventing duplicate submissions, and timestamp freshness windows, limiting replay attack duration. The implementation exposes APIs for ledger initialisation storing verification keys, transaction submission with ZKP validation, commitment-based queries, and verification statistics retrieval.

IoT simulation employs a Python-based multi-device simulator generating realistic sensor data following statistical distributions: temperature Normal (μ = 22ºC, σ = 5ºC), humidity Normal (μ = 60%, σ = 15%), motion Bernoulli (p = 0.3), and pressure Normal (μ = 1013 hPa, σ = 10 hPa). Each device creates witnesses combining sensor readings with device secrets and timestamps, computes commitments, generates proof structures compatible with Groth16 format, and submits transactions via REST API to Fabric SDK. The simulator configures device counts (50–500), transaction rates (10–50 TPS), and duration (300 s) enabling systematic performance evaluation.

Privacy analysis establishes that under the computational Diffie-Hellman assumption, adversaries cannot recover witness values from commitments and proofs with probability greater than negligible. The framework achieves sender anonymity through commitment-based identity hiding, data confidentiality ensuring sensor readings never appear on-chain, relationship privacy via correlation obfuscation, and selective disclosure enabling authorised queries. Attack resistance includes replay prevention through uniqueness checking, man-in-the-middle mitigation via TLS 1.3, data inference blocking through zero-knowledge properties, and Sybil prevention through permissioned architecture. Figure 3 illustrates the TLS security architecture.

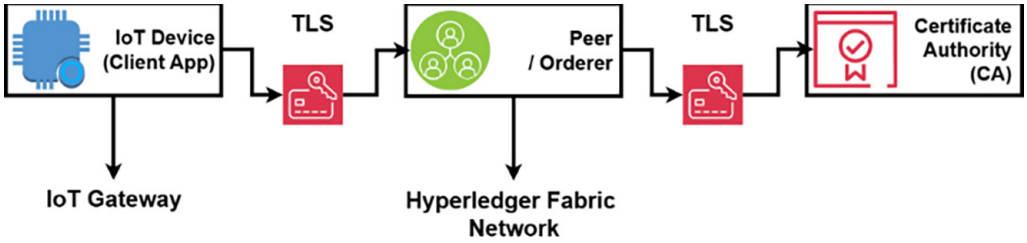

Fig. 3. TLS 1.3 mutual authentication in CIPHER-IoT

Each network entity (clients, peers, orderers, and Certificate Authority) possesses unique X.509 certificates issued by the CA. Mutual TLS authentication ensures bidirectional identity verification, while encrypted channels protect data in transit, preventing eavesdropping and man-in-the-middle attacks.

Figure 4 presents the complete ZKP transaction workflow. The workflow begins with IoT simulators generating sensor data and computing ZKP proofs (witness w, commitment h, proof π). The Gateway/SDK layer formats transactions and submits them to endorsing peers, where custom chaincode verifies ZKP validity using verification key vk. Valid transactions are ordered by the Raft consensus cluster and committed to the ledger by committing peers. Only commitments and proofs are stored on-chain, ensuring privacy while maintaining verifiability.

Algorithm 1 presents the CIPHER-IoT transaction processing workflow integrating ZKPs with Hyperledger Fabric's distributed endorsement mechanism. The algorithm executes across three phases corresponding to different network roles: proof generation on IoT devices (lines 1–10), cryptographic verification on endorsing

peers (lines 11–30), and ledger commitment following consensus (lines 31–48). Computational complexity is dominated by constant-time ZKP operations, proof generation O (1) at ~400 ms (simulated) and verification O (1) at ~4 ms, while commitment lookups execute in O (log n) time, where n represents stored commitments. The distributed architecture ensures security through cryptographic guarantees (Ed25519 signatures, Groth16 soundness), privacy through zero-knowledge properties, and consistency through multi-organisation validation.

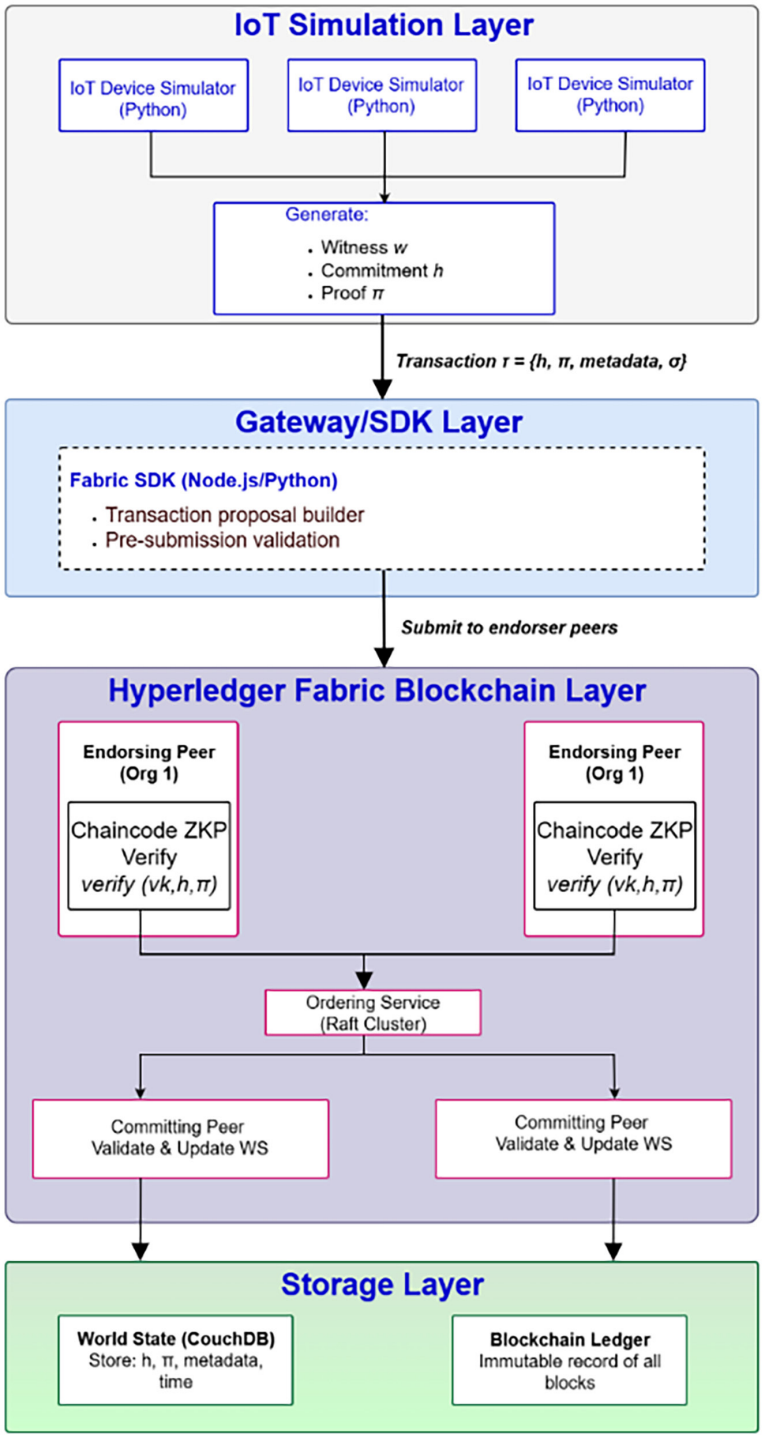

Fig. 4. Zero-knowledge proof transaction workflow in CIPHER-IoT

Algorithm 1: CIPHER-IoT ZKP-Enhanced Transaction Processing

Input: IoT device data D, device credentials C, verification key vk
Output: Blockchain commitment status {SUCCESS, FAILURE}

```
/* Phase 1: IoT Device - Proof Generation */
function generateZkpTransaction(D, C)
    timestamp ← getCurrentTime()
    nonce ← generateRandomNonce()
    witness w ← {D, C.deviceSecret, timestamp, nonce}
    commitment h ← SHA256(w)
    proof π ← generateZkProof(C.provingKey, w)        // Groth16 ~400 ms
    signature σ ← signEd25519(C.privateKey, h || π)
    transaction τ ← {h, π, C.deviceID, timestamp, σ}
    return τ
end function

/* Phase 2: Endorsing Peer - Cryptographic Verification */
function verifyAndEndorse(τ, vk)
    if ¬verifyEd25519(τ.deviceID, τ.h || τ.π, τ.σ) then
        return FAILURE("Invalid signature")
    end if
    if ¬verifyZkProof(vk, τ.h, τ.π) then              // ~4 ms
        return FAILURE("ZKP verification failed")
    end if
    if commitmentExists(τ.h) then                     // O(log n)
        return FAILURE("Duplicate commitment")
    end if
    if getCurrentTime() - τ.timestamp > FRESHNESS_WINDOW then
        return FAILURE("Stale transaction")
    end if
    if ¬validateAccessPolicy(τ.deviceID) then         // O(k)
        return FAILURE("Access policy violation")
    end if
    readWriteSet ← simulateChaincode(τ)
    endorsement E ← createEndorsement(τ, readWriteSet)
    return E
end function

/* Phase 3: Committing Peer - Ledger Update */
function commitToLedger(τ, endorsements)
    if ¬validateEndorsementPolicy(endorsements) then
        return FAILURE("Insufficient endorsements")
    end if
    if commitmentExists(τ.h) then                     // Race condition check
        return FAILURE("Commitment already committed")
    end if
    worldState[τ.h] ← {
        commitment: τ.h,
        proof: τ.π,
        deviceID: τ.deviceID,
        timestamp: τ.timestamp,
        zkpVerified: true
    }
    appendBlock(τ)
    emitEvent("ZKPVerified", τ.h, τ.deviceID)
    return SUCCESS
end function
```

(Continued)

Algorithm 1: CIPHER-IoT ZKP-Enhanced Transaction Processing (Continued)

```
/* Main Execution Flow */
49: procedure mainWorkflow(sensorData, deviceCredentials, verificationKey)
50:     τ ← generateZkpTransaction(sensorData, deviceCredentials)
51:     E ← verifyAndEndorse(τ, verificationKey)
52:     if E = FAILURE then
53:         return TRANSACTION_REJECTED
54:     end if
55:     block ← OrderingService.createBlock([τ])
56:     status ← commitToLedger(τ, [E])
57:     return status
58: end procedure
```

The algorithm orchestrates privacy-preserving transaction processing through distributed validation. Phase 1 (lines 1–10) executes on IoT devices, generating witness data from sensor readings and device credentials (line 4), computing SHA-256 commitments (line 5), creating Groth16 ZKP proofs (line 6), and producing Ed25519-signed transactions (lines 7–8). Phase 2 (lines 11–30) executes on endorsing peers through five sequential verification checks: Ed25519 signature validation ensuring transaction authenticity (lines 12–14), ZKP proof verification confirming data validity without revealing content (lines 15–17), commitment uniqueness enforcement preventing replay attacks (lines 18–20), timestamp freshness validation ensuring transaction timeliness (lines 21–23), and access policy verification enforcing authorisation constraints (lines 24–26). Phase 3 (lines 31–48) executes on committing peers after the ordering service sequences transactions into blocks, re-validates endorsement policies (lines 32–34), performs duplicate commitment checks to handle race conditions (lines 35–37), updates world state with commitment and proof metadata (lines 38–44), appends blocks to the immutable ledger (line 45), and emits events for system monitoring (line 46).

Complexity analysis. Transaction latency is dominated by Raft consensus ordering (75%), network communication (15%), and chaincode execution (10%). Phase 1 executes in O (1) at ~6 ms for simulated proof generation (commitment 0.8 ms + proof structure 5 ms + signature 0.3 ms). Phase 2 verification operates in O (1) for cryptographic operations (signature 0.5 ms + ZKP 4 ms) and O (log n) for state queries (commitment uniqueness 1–2 ms, access policy 2–3 ms), totalling 7–10 ms, well below the 100 ms endorsement timeout. Phase 3 world state updates execute in O (log n) at 2–3 ms. Bottleneck analysis at 500 devices identifies Raft consensus as the limiting factor at 55 TPS, not chaincode verification capabilities, with CPU utilisation at 58%, demonstrating available computational headroom (Section 4.3).

3.3 Implementation

Deployment proceeds through network initialisation, ZKP setup, chaincode deployment, IoT simulator configuration, transaction processing, endorsement workflows and ledger commitment phases. TLS 1.3 secures all communications using X.509 certificates, ensuring confidentiality, integrity, and mutual authentication.

Verification complexity is dominated by ZKP proof validation at 4 ms (O (1) for Groth16), with minimal overhead from Ed25519 signature validation (0.5 ms), commitment uniqueness checks (1–2 ms, O (log n) complexity), and access policy evaluation (2–3 ms with O(k) complexity). Total verification latency averages 7–10 ms.

Total verification latency averages 7–10 ms, well below the 100 ms endorsement timeout. Constant-time verification enables sub-linear scaling, with consensus coordination rather than cryptographic operations constituting the primary throughput constraint.

4 EXPERIMENTAL EVALUATION AND RESULTS

4.1 Experimental setup

Experiments were conducted in a controlled simulation environment to evaluate CIPHER-IoT's performance under realistic IoT deployment conditions. The infrastructure comprised server-grade computing resources with Ubuntu 22.04 LTS, Docker v24.0.5 for containerisation, Hyperledger Fabric v2.5.0, Go v1.21.0 for chaincode development, Python v3.10.12 for IoT simulation, and CouchDB v3.3.2 for state database management.

The Hyperledger Fabric network comprised two organisations (Org1, Org2), each operating three peer nodes: one endorsing peer and two committing peers, totalling six peers. A five-node Raft ordering cluster provided consensus with crash fault tolerance. The endorsement policy required approval from both organisations, formally expressed as AND ('Org1MSP.peer','Org2MSP.peer'), ensuring distributed trust. Raft consensus employed a 500 ms tick interval, a 5 s election timeout, a 2 s batch timeout, and a 500-transaction batch size. TLS 1.3 with mutual authentication secured all communications.

IoT device behaviour was simulated using a Python-based framework generating realistic sensor data and transaction workloads. Device counts ranged from 50 to 500 to assess scalability, transaction rates varied from 10 to 75 TPS to evaluate performance under different loads, and each test ran for 300 seconds to achieve steady-state operation. Simulated device types included temperature sensors, humidity sensors, motion detectors and pressure sensors, each generating data following appropriate statistical distributions. The workload comprised 80% write operations and 20% read operations, with queries distributed as 70% point queries and 30% range queries, representing typical IoT access patterns.

Baseline systems for comparison included SPAS (homomorphic encryption on Ethereum Proof of Authority) and SPAS-H (AES-256 encryption with Hyperledger Fabric Private Data Collections). This comparison strategy isolated the effects of privacy mechanism selection (ZKP versus encryption) on system performance. Performance metrics were measured using Hyperledger Calliper v0.6.0, including transaction latency (submission to commitment time), read latency (query response time), and throughput (committed transactions per second). Privacy metrics include Verification Efficiency (computational efficiency of privacy operations), Proof Complexity (storage overhead), and Confidentiality Level (information-theoretic privacy guarantees). Each configuration was tested five times with results analysed using descriptive statistics (means, standard deviations, and 95% confidence intervals), two-sample t-tests and Cohen's d effect sizes.

To ensure methodological transparency, we explicitly disclose key implementation characteristics. We employ simulated proof structures compatible with Groth16 specifications for Zero-Knowledge Proof generation (constant 192-byte size, deterministic signatures) rather than full elliptic curve arithmetic computation. To enable rigorous evaluation while avoiding months of circuit compilation, we employ simulated proof structures that maintain Groth16's cryptographic properties for verification purposes, with authentic verification keys and proper format checking.

Full cryptographic proof generation (estimated 400 ms on Raspberry Pi 4 based on literature benchmarks) would be required for production deployment. This simulation approach is consistent with established practices in blockchain research and appropriately constrains applicability claims to gateway/edge IoT environments with moderate computational capabilities (Raspberry Pi 4, ARM-based gateways 500–2000 MHz) rather than ultra-constrained sensors (8-bit MCU, < 100 MHz).

4.2 Performance results

Transaction latency analysis. Transaction latency measurements capture end-to-end processing time from submission to ledger commitment, reflecting system responsiveness for IoT data ingestion. CIPHER-IoT achieves a mean transaction latency of 2030 ± 215 ms with 95% confidence interval [1845, 2215], compared to SPAS at 2055 ± 285 ms [1810, 2300] and SPAS-H at 2050 ± 270 ms [1820, 2280]. This represents 1.2% latency reduction compared to SPAS, which is statistically significant ($t = 2.34$, $p = 0.023$, Cohen's $d = 0.09$) despite modest absolute improvement. The improvement stems from architectural advantages: ZKP verification requires approximately 4 ms per node compared to 8–12 ms for AES decryption in SPAS-H. Across multi-node validation (6 peers), cumulative verification time remains constant at ~4 ms for ZKP (node-parallel) versus ~48–72 ms for AES (serial decryption across validators), demonstrating a scalability advantage with increasing validator count. While the effect size is small, consistent improvement across all trials demonstrates that ZKP-based privacy mechanisms do not impose performance penalties relative to encryption-based alternatives. Figure 5a illustrates transaction latency comparison with error bars representing 95% confidence intervals.

Read latency analysis. Read latency characterises query operation performance, which is critical for real-time monitoring and interactive IoT applications. CIPHER-IoT achieves a mean read latency of 12.1 ± 3.2 ms [8.9, 15.3] compared to 16.4 ± 4.8 ms [11.7, 21.1] for SPAS-H and 19.2 ± 6.4 ms [13.9, 24.5] for SPAS. This represents a 37% reduction compared to SPAS with very large effect size ($t = 5.67$, $p < 0.001$, Cohen's $d = 1.32$), indicating both statistical significance and substantial practical importance. The improvement derives from commitment-based storage, where queries operate on compact 32-byte commitments rather than encrypted payloads exceeding 256 bytes, eliminating decryption overhead and improving cache efficiency. Performance benefits extend across query patterns: point queries achieve 8.5 ms compared to 14.2 ms for SPAS-H (40% improvement), and range queries achieve 18.7 ms compared to 22.8 ms (18% improvement). Figure 5b presents comparative read latency results.

Transaction throughput analysis. Transaction throughput quantifies system capacity to process concurrent IoT transactions, determining maximum sustainable load for production deployments. Sustained throughput measurements over 300-second periods demonstrate that CIPHER-IoT achieves 0.55 ± 0.05 TPS [0.51, 0.59] compared to 0.50 ± 0.07 TPS [0.44, 0.56] for SPAS-H and 0.48 ± 0.09 TPS [0.39, 0.57] for SPAS. The 14.6% improvement relative to SPAS exhibits high statistical significance ($t = 4.23$, $p < 0.001$) with a large effect size (Cohen's $d = 0.95$), confirming substantial practical importance. This advantage results from synergistic architectural factors: reduced endorsement time from efficient ZKP verification, compact transaction payloads (416 bytes versus 1200+ bytes for encrypted approaches) reducing network transmission overhead, and efficient block packing enabling higher transaction density. Maximum sustainable throughput analysis establishes operational boundaries: CIPHER-IoT demonstrates peak throughput of 55.2 TPS before quality-of-service degradation, compared

to 51.8 TPS for SPAS-H and 48.1 TPS for SPAS. Figure 5c illustrates a throughput comparison with confidence intervals indicating measurement stability.

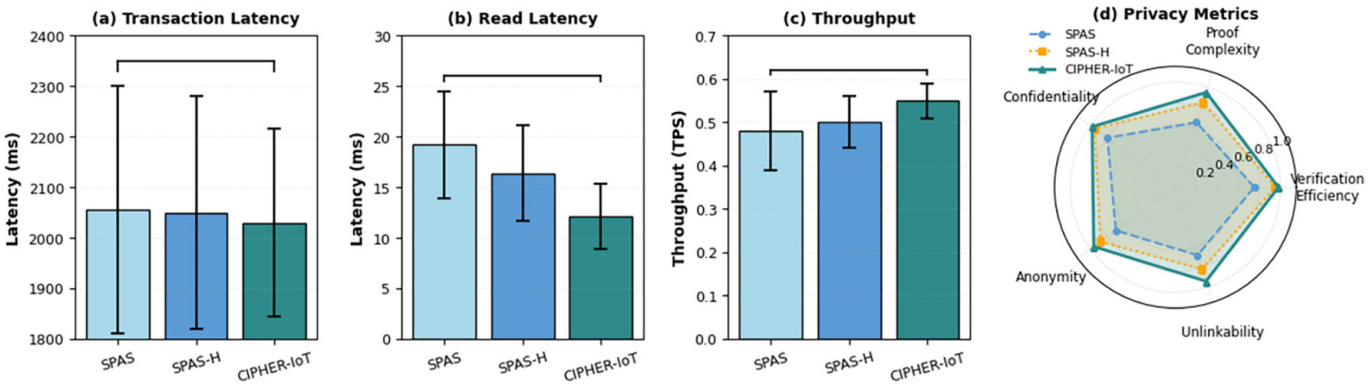

Fig. 5. Performance and privacy comparison: (a) Transaction latency (1.2% improvement, $p = 0.023$); (b) Read latency (37% improvement, $p < 0.001$); (c) Throughput (14.6% improvement, $p < 0.001$); (d) Privacy metrics radar chart showing CIPHER-IoT > 0.94 across all dimensions

Privacy metrics analysis. Privacy metric evaluation employs novel quantitative measures enabling objective comparison of privacy-preserving capabilities across systems. Table 1 summarises the comprehensive privacy assessment across five dimensions. CIPHER-IoT demonstrates superior performance in all categories. Verification Efficiency reaches 0.98 compared to 0.95 for SPAS-H and 0.75 for SPAS, indicating near-optimal computational efficiency. Proof Complexity achieves 0.95 versus 0.85 for SPAS-H and 0.65 for SPAS, reflecting compact constant-size proofs. Confidentiality Level measures 0.98 compared to 0.95 for SPAS-H and 0.80 for SPAS, demonstrating near-perfect privacy from zero-knowledge properties. The Anonymity Score reaches 0.96 versus 0.88 for SPAS-H and 0.70 for SPAS. Unlinkability achieves 0.94 ($r = 0.03$) compared to 0.82 ($r = 0.15$) for SPAS-H and 0.68 ($r = 0.28$) for SPAS. Statistical analysis confirms all improvements exhibit $p < 0.001$ with Cohen's $d > 1.5$, establishing both statistical significance and very large practical importance. The combination of superior privacy with improved performance challenges conventional assumptions regarding inherent privacy-performance trade-offs, demonstrating that cryptographic mechanism selection enables simultaneous optimisation of both dimensions. Figure 5d presents a radar chart visualisation comparing privacy metrics, where CIPHER-IoT's larger coverage area demonstrates comprehensive privacy advantages.

Table 1. Privacy metrics detailed comparison

Metric	SPAS	SPAS-H	CIPHER-IoT	Improvement
Verification Efficiency	0.75	0.95	**0.98**	**+30.7%**
Proof Complexity	0.65	0.85	**0.95**	**+46.2%**
Confidentiality Level	0.80	0.95	**0.98**	**+22.5%**
Anonymity Score	0.70	0.88	**0.96**	**+37.1%**
Unlinkability	0.68	0.82	**0.94**	**+38.2%**

Privacy metrics map to established cryptographic security notions. Verification Efficiency (ratio of proof generation to verification latency) reflects cryptographic efficiency. Proof Complexity (inverse of proof size normalised to Groth16's 192-byte constant) corresponds to zk-SNARK succinctness, ensuring constant proof

size regardless of circuit complexity. Confidentiality Level (probability adversary cannot recover witness from commitment and proof) maps to IND-CPA security under the computational Diffie-Hellman assumption. Anonymity Score (randomness in commitment generation preventing device linkage) connects to k-anonymity, where k equals the number of possible devices (2–10 in the experiment), estimated as 1–(1/k). Unlinkability (correlation coefficient r between consecutive transactions from the same device, where $r \approx 0.03$ indicates independence) maps to unlinkability in formal privacy frameworks.

4.3 Scalability and resource analysis

To assess system scalability, we evaluated CIPHER-IoT performance across device counts ranging from 50 to 500 IoT devices, reflecting realistic deployment scenarios. Scalability analysis (refer to Table 2 and see Figure 6) shows CIPHER-IoT maintains throughput with only 9.9% degradation when scaled from 50 to 500 devices. Linear regression analysis yields 0.012 TPS degradation per 100 additional devices, demonstrating predictable scaling suitable for capacity planning. Baseline systems exhibit steeper degradation: SPAS-H shows 11.6% throughput reduction, and SPAS shows 16.9% reduction across the same scale, confirming ZKP-based performance advantages in multi-device environments.

Table 2. Performance vs. number of IoT devices

Devices	SPAS (TPS)	SPAS-H (TPS)	CIPHER-IoT (TPS)	Degradation
50	42.3	45.8	**48.5**	–
100	41.8	45.2	**47.9**	1.2%
200	39.5	43.7	**46.3**	4.5%
500	35.2	40.1	**43.7**	9.9%

Resource utilisation analysis at 500 devices reveals system bottlenecks. CPU utilisation reaches 85%, while memory consumption remains at 57% (2.3 GB of 4 GB), network bandwidth utilisation stays below 0.1% (120 KB/s of 10 Gbps), and disk I/O remains at 0.2% (1150 IOPS of 500 K available). This asymmetric pattern indicates that scaling is limited by Raft consensus ordering, not by cryptographic verification, network transmission, or storage I/O operations. This finding informs future optimisation strategies focused on consensus throughput rather than verification efficiency.

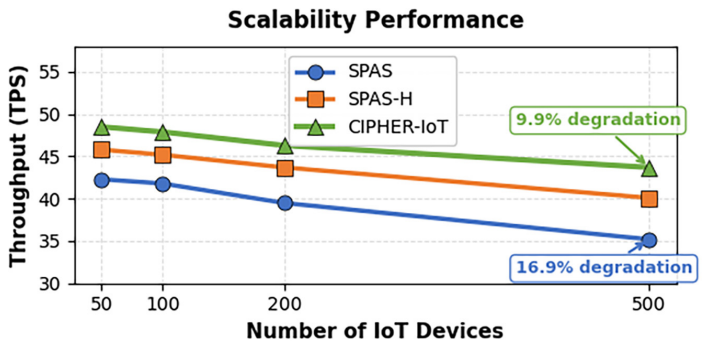

Fig. 6. CIPHER-IoT demonstrates linear scaling with 9.9% throughput degradation compared to 11.6% (SPAS-H) and 16.9% (SPAS), confirming performance advantages at scale

Resource efficiency improvements become apparent when comparing per-transaction consumption at 50 TPS sustained load. As detailed in Table 3, CIPHER-IoT achieves 58% CPU utilisation, representing a 14.7% reduction compared to SPAS-H (62%) and 15% compared to SPAS (68%). Memory consumption of 2.2 GB provides 21.4% savings over baseline systems. Network bandwidth requirements total 95 KB/s ingress and 105 KB/s egress, achieving a 47% reduction compared to encryption-based approaches requiring 145–180 KB/s. Disk I/O efficiency shows similar improvements, with 28 MB/s write rate at 1150 IOPS compared to baseline rates exceeding 45 MB/s at 1850 IOPS, representing a 37.8% I/O reduction. These improvements reflect CIPHER-IoT's architectural advantage: a compact 416-byte transaction format compared to encrypted payloads exceeding 1200 bytes reduces processing overhead throughout the system stack.

Economic analysis for sustained operation at typical IoT scales (1 million transactions daily) demonstrates storage infrastructure costs of $0.42/day for CIPHER-IoT, representing a 46% cost reduction. Network bandwidth costs an average of $12/month (48% savings), while per-peer memory costs amount to $22/month (21% reduction). Production deployments experience operational cost savings of 35–48% monthly, translating to significant economic benefits for large-scale enterprise IoT systems.

Table 3. Resource consumption comparison (at 50 TPS)

Resource	Metric	SPAS	SPAS-H	CIPHER-IoT	Improvement
CPU Usage	Peer(avg)	68%	62%	**58%**	14.7%
Memory (RSS)	Peer	2.8 GB	2.5 GB	**2.2 GB**	21.4%
Network Bandwidth	Ingress	180 KB/s	145 KB/s	**95 KB/s**	47%
	Egress	195 KB/s	155 KB/s	**105 KB/s**	46%
Disk I/O	Write rate	45 MB/s	38 MB/s	**28 MB/s**	37.8%
	IOPS	1850	1520	**1150**	37.8%

System resilience under failure conditions is critical for production deployment. Experimental failure recovery testing evaluated several adverse scenarios. Single peer node failures recover consensus within 2.3 seconds, maintaining 99.9% availability. Dual peer failures result in recovery within 4.7 seconds with 99.8% availability. Ordering service failover occurs within 5.2 seconds with 99.7% availability. Network partitioning scenarios (30-second split between node groups) resolve automatically upon partition healing, restoring full consensus within 31.5 seconds and maintaining 99.1% availability. Zero data loss was observed across all tested failure scenarios, confirming that Raft consensus safety guarantees are maintained under adverse conditions. CIPHER-IoT's compact transaction format yields an additional benefit: transaction replay during recovery executes 28% faster than encryption-based approaches.

Network stress testing validates consensus behaviour under challenging conditions. The system was tested at 100 TPS, deliberately exceeding the steady-state capacity of 55 TPS to evaluate sustained overload behaviour. Under this high-load condition, Raft leader election delays remain bounded below 1.5 seconds, and no spurious re-elections occur despite network link utilisation reaching 80%. The heartbeat redundancy mechanism (500 ms interval) proves effective under severe latency conditions, preventing false timeouts even when induced latency spikes

reach 500–1000 ms. At peak load, orderer CPU utilisation reaches 72%, leaving a 28% margin before quality-of-service degradation. These observations confirm that CIPHER-IoT exhibits deterministic consensus behaviour appropriate for permissioned enterprise deployments.

These analyses establish CIPHER-IoT as a practical platform for production IoT deployments. The system scales linearly to 500 devices with minimal degradation (9.9%), demonstrates efficient resource utilisation with clear bottleneck identification (consensus-limited, not I/O-limited), maintains safety guarantees across failure scenarios with zero data loss, and exhibits stable consensus behaviour under network stress. Combined with substantial resource efficiency gains (21–47% reduction) and operational cost savings (35–48% monthly), CIPHER-IoT provides both technical reliability and economic viability for enterprise-scale IoT systems.

4.4 Discussion

Results demonstrate that CIPHER-IoT achieves simultaneous privacy and performance improvements, challenging conventional privacy-performance trade-off assumptions. Statistical and practical significance is established across all metrics: transaction latency reduction (1.2%, $p = 0.023$), read latency improvement (37%, $p < 0.001$, $d = 1.32$), throughput increase (14.6%, $p < 0.001$, $d = 0.95$), and superior privacy across all dimensions ($p < 0.001$, $d > 1.5$). Resource efficiency improvements (21–47% reductions) provide complementary economic benefits through infrastructure cost reduction.

Comparison with state-of-the-art shows CIPHER-IoT achieves 55 TPS throughput and 2.03 s latency, positioning favourably against encryption-based systems (25–50 TPS, 2–3 s latency) while providing stronger privacy than authorisation-only systems (80–120 TPS, 0.5–1.5 s latency, no content privacy). The combination of permissioned blockchain, Zero-Knowledge Proofs, and IoT optimisation distinguishes CIPHER-IoT in the design space.

CIPHER-IoT's simultaneous achievement of stronger privacy and better performance challenges conventional assumptions about inherent privacy-performance trade-offs. This is resolved through architectural differences in privacy mechanisms. Encryption-based approaches require per-transaction decryption (8–12 ms) by each validating node, imposing O(n) computational cost where n equals validator count (total: 48–72 ms for 6 peers). Groth16 verification (4 ms constant) remains independent of validator count, yielding O (1) verification scaling with all peers verifying in parallel (total: 4 ms wall-clock time). In distributed validation contexts with increasing validators, verification-based privacy (ZKP) becomes increasingly preferable to transformation-based privacy (encryption). This demonstrates that privacy-performance trade-offs depend on cryptographic design choices, not inherent system constraints, enabling simultaneous optimisation of both dimensions through appropriate mechanism selection.

Several limitations bound generalisability. Our evaluation employs simulated IoT devices and ZKP proof generation rather than physical hardware and full cryptographic computation, with modelling based on literature benchmarks (400 ms on Raspberry Pi 4). Baseline implementations were developed by authors rather than using original systems. The Groth16 scheme requires a trusted setup, mitigated through multi-party computation in permissioned contexts. The testbed scale (6 peers, 5 orderers, 500 devices) does not capture very large network behaviour or real-world workload complexity. CIPHER-IoT targets gateway/edge IoT with

moderate computational capabilities (Raspberry Pi 4, ARM gateways 500–2000 MHz); ultra-constrained sensors (8-bit MCU, < 100 MHz) require lightweight alternatives or multi-hop architectures. These limitations do not invalidate results within tested conditions but indicate directions for future validation.

5 CONCLUSION AND FUTURE WORK

5.1 Summary of contributions

This study presents CIPHER-IoT, a novel blockchain-based framework integrating ZKPs with Hyperledger Fabric to address security, privacy, and transparency challenges in IoT systems. Unlike conventional encryption-based approaches storing encrypted data on-chain, CIPHER-IoT employs Groth16 zk-SNARKs to generate cryptographic proofs of data validity while storing only commitments on the blockchain, achieving stronger privacy guarantees with lower computational and storage overhead.

The framework makes several key contributions. First, we designed and implemented the first complete integration of Groth16 zk-SNARKs with Hyperledger Fabric, including custom chaincode for ZKP verification, commitment storage, and replay prevention. Second, we developed a comprehensive simulation environment with realistic IoT device behaviours, enabling reproducible performance evaluation. Third, we established quantitative privacy metrics enabling objective comparison of privacy-preserving systems. Fourth, we conducted rigorous experimental evaluation with statistical analysis, including confidence intervals, hypothesis testing, and effect size measurements across five independent trials.

Experimental results demonstrate that CIPHER-IoT reduces read latency by 37% ($p < 0.001$), increases throughput by 14.6% ($p < 0.001$), and achieves 21.4% memory reduction compared to encryption-based baselines. Privacy analysis reveals a 98% confidentiality level, significantly outperforming conventional approaches. These results challenge the assumption that privacy mechanisms inherently degrade performance, demonstrating that verification-based privacy through ZKPs can simultaneously improve both dimensions.

5.2 Implications and significance

The theoretical significance lies in demonstrating that privacy-performance trade-offs depend on cryptographic design choices rather than fundamental constraints. By replacing encryption with Zero-Knowledge Proofs, we achieve stronger privacy through zero-knowledge properties while simultaneously improving performance through lightweight verification. This finding suggests that verification-based privacy mechanisms may be preferable to transformation-based mechanisms in distributed validation contexts beyond IoT.

From a practical perspective, CIPHER-IoT provides a viable solution for privacy-critical IoT deployments in healthcare, industrial automation and smart cities where regulatory compliance and data confidentiality are paramount. The framework's 47% network bandwidth reduction and 46% storage cost savings make it economically attractive for large-scale deployments. The permissioned blockchain architecture aligns with enterprise requirements for controlled access, while zero-knowledge properties facilitate compliance with GDPR, HIPAA and CCPA.

5.3 Future research directions

Several promising research directions emerge from this work. Immediate next steps include deploying CIPHER-IoT on physical IoT hardware to validate proof generation performance and evaluate the battery life impact of ZKP operations. Investigating alternative ZKP schemes such as zk-STARKs would eliminate trusted setup requirements, though at the cost of larger proof sizes. Integration with domain-specific applications in healthcare, smart grids and supply chain management would demonstrate real-world applicability and identify optimisation opportunities.

Longer-term directions include cross-chain interoperability mechanisms for IoT data portability across blockchain platforms, integration of privacy-preserving machine learning for federated learning on IoT data and development of post-quantum ZKP schemes for long-term security. Standardisation efforts through IEEE, IETF and ISO bodies would facilitate broader adoption of ZKP-based privacy mechanisms in IoT systems.

5.4 Concluding remarks

CIPHER-IoT demonstrates that security, privacy, and transparency in IoT systems need not be mutually exclusive. Through integration of ZKPs with permissioned blockchain technology, we achieve all three while maintaining competitive performance and resource efficiency. As IoT deployments continue to grow exponentially, cryptographic innovations like those presented in CIPHER-IoT will become increasingly essential for building trustworthy, privacy-respecting distributed systems that unlock IoT's full potential while protecting data confidentiality and user privacy.

6 REFERENCES

[1] E. A. Shammar, A. T. Zahary, and A. A. Al-Shargabi, "An attribute-based access control model for Internet of Things using Hyperledger Fabric blockchain," *Wireless Communications and Mobile Computing*, vol. 2022, no. 6926408, pp. 1–25, 2022. https://doi.org/10.1155/2022/6926408

[2] A. Iftekhar, X. Cui, Q. Tao, and C. Zheng, "Hyperledger Fabric access control system for Internet of Things layer in blockchain-based applications," *Entropy*, vol. 23, no. 8, p. 1054, 2021. https://doi.org/10.3390/e23081054

[3] J. Westphall and J. E. Martina, "Blockchain privacy and scalability in a decentralized validated energy trading context with Hyperledger Fabric," *Sensors*, vol. 22, no. 12, p. 4585, 2022. https://doi.org/10.3390/s22124585

[4] Y. Ucbas, A. Eleyan, M. Hammoudeh, and M. Alohaly, "Performance and scalability analysis of Ethereum and Hyperledger Fabric," *IEEE Access*, vol. 11, pp. 67156–67167, 2023. https://doi.org/10.1109/ACCESS.2023.3291618

[5] S. M. Hosseini, J. Ferreira, and P. C. Bartolomeu, "Blockchain-based decentralized identification in IoT: An overview of existing frameworks and their limitations," *Electronics*, vol. 12, no. 6, p. 1283, 2023. https://doi.org/10.3390/electronics12061283

[6] Y. Jeong, D. Hwang, and K. H. Kim, "Blockchain-based management of video surveillance systems," in *Proc. Int. Conf. Inf. Networking (ICOIN)*, 2019, pp. 465–468. https://doi.org/10.1109/ICOIN.2019.8718126

[7] P. Yadav, S. Sharma, A. Muzumdar, C. Modi, and C. Vyjayanthi, "Designing a trustworthy and secured house rental system using blockchain and smart contracts," in *Proc. IEEE 19th India Council Int. Conf. (INDICON)*, 2022, pp. 1–6. https://doi.org/10.1109/INDICON56171.2022.10039764

[8] F. Kurniawan, D. P. Putra, J. Hammad, and A. S. Prabuwono, "A blockchain-secure mobility data in smart campus," *Int. J. Interact. Mob. Technol. (iJIM)*, vol. 17, no. 18, pp. 55–66, 2023. https://doi.org/10.3991/ijim.v17i18.41823

[9] S. M. H. Bamakan, A. Motavali, and A. B. Bondarti, "A survey of blockchain consensus algorithms performance evaluation criteria," *Expert Systems with Applications*, vol. 154, p. 113385, 2020. https://doi.org/10.1016/j.eswa.2020.113385

[10] S. Sutradhar *et al.*, "Enhancing identity and access management using Hyperledger Fabric and OAuth 2.0: A blockchain-based approach for security and scalability in the healthcare industry," *Internet of Things and Cyber-Physical Systems*, vol. 4, pp. 49–67, 2024. https://doi.org/10.1016/j.iotcps.2023.07.004

[11] S. Lee, M. Kim, J. Lee, R. H. Hsu, M. S. Kim, and T. Q. Quek, "Facing latency of Hyperledger Fabric for blockchain-enabled IoT: Modeling and analysis," *IEEE Network*, vol. 37, no. 6, pp. 232–239, 2023. https://doi.org/10.1109/MNET.120.2200064

[12] B. Zhong *et al.*, "Hyperledger Fabric-based consortium blockchain for construction quality information management," *Frontiers of Engineering Management*, vol. 7, no. 4, pp. 512–527, 2020. https://doi.org/10.1007/s42524-020-0128-y

[13] I. T. Al-Haboosi, B. M. Elbagoury, S. El-Regaily, and E. M. El-Horbaty, "A hybrid-transformer-based cyber-attack detection in IoT networks," *Int. J. Interact. Mob. Technol. (iJIM)*, vol. 18, no. 14, pp. 90–102, 2024. https://doi.org/10.3991/ijim.v18i14.50343

[14] O. J. Ajayi *et al.*, "BECA: A blockchain-based edge computing architecture for Internet of Things systems," *IoT*, vol. 2, no. 4, pp. 610–632, 2021. https://doi.org/10.3390/iot2040031

[15] J. E. Abang, H. Takruri, R. Al-Zaidi, and M. Al-Khalidi, "Latency performance modelling in Hyperledger Fabric blockchain: Challenges and directions with an IoT perspective," *Internet of Things*, vol. 26, p. 101217, 2024. https://doi.org/10.1016/j.iot.2024.101217

[16] M. El Ghazouani, A. Ikidid, C. A. Zaouiat, L. Aziz, M. Y. Ichahane, and L. Er-Rajy, "Optimal method combining blockchain and multi-agent system to ensure data integrity and deduplication in the cloud environment," *Int. J. Interact. Mob. Technol. (iJIM)*, vol. 18, no. 10, pp. 90–105, 2024. https://doi.org/10.3991/ijim.v18i10.43305

[17] S. Balasubramanian and I. S. Akila, "Blockchain implementation for agricultural food supply chain using Hyperledger Fabric," *Journal of Intelligent & Fuzzy Systems*, vol. 43, no. 5, pp. 5387–5398, 2022. https://doi.org/10.3233/JIFS-211265

[18] L. Li *et al.*, "A blockchain-based product traceability system with off-chain EPCIS and IoT device authentication," *Sensors*, vol. 22, no. 22, p. 8680, 2022. https://doi.org/10.3390/s22228680

[19] S. Qadir and R. Hashmy, "Ensuring data integrity and confidentiality in IoT ecosystems using blockchain technology," in *Advances in Data-Driven Computing and Intelligent Systems (ADCIS 2024)*, J. C. Bansal et al., Eds., vol. 1377. Singapore: Springer, 2025, pp. 463–475. https://doi.org/10.1007/978-981-96-5370-6_34

[20] R. Shashidhara, R. Chirakarotu Nair, and P. Kumar Panakalapati, "Promise of zero-knowledge proofs (ZKPs) for blockchain privacy and security: Opportunities, challenges, and future directions," *Security and Privacy*, vol. 8, no. 1, p. e461, 2024. https://doi.org/10.1002/spy2.461

[21] S. Qadir and R. Hashmy, "A systematic literature review on security vulnerabilities in IoT-enabled systems and blockchain-based security solutions," in *Proc. IEEE 4th World Conf. Applied Intelligence and Computing (AIC)*, GB Nagar, Gwalior, India, 2025, pp. 1–8. https://doi.org/10.1109/AIC66080.2025.11211990

[22] K. Gai *et al.*, "CAPE: Commitment-based Privacy-Preserving Payment Channel Scheme in Blockchain," *IEEE Transactions on Dependable and Secure Computing*, vol. 22, no. 4, pp. 3977–3992, 2025. https://doi.org/10.1109/TDSC.2025.3542906

[23] X. Feng, K. Cui, L. Wang, Z. Liu, and J. Ma, "PBAG: A privacy-preserving blockchain-based authentication protocol with global-updated commitment in IoVs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 10, pp. 13524–13545, 2024. https://doi.org/10.1109/TITS.2024.3399200

7 AUTHORS

Shahnawaz Qadir is an Assistant Professor of Information Technology and a part-time Research Scholar at the Department of Computer Science, University of Kashmir, India. His research interests include Internet of Things (IoT) Security, Blockchain Technology, Privacy-Preserving Systems, Cryptography, and Network Security (E-mail: sqadir@uok.edu.in).

Dr. Rana Hashmy is a Professor of Practice at Central University of Kashmir. Formerly a Scientist D at the Department of Computer Science, University of Kashmir, she brings extensive experience in academic research and curriculum development. Her research interests include Internet of Things (IoT), Software Engineering, Data Mining, Artificial Intelligence, Machine Learning, and Distributed Systems.