BISF-IoT: A Scalable Blockchain-Integrated Security Framework with Formal Guarantees

Shahid Imran[1], Kalsoom Safdar[1,2*], Muhammad Usman Younus[3,4*]
[1]Department of Computer Science & IT, University of Jhang, Jhang, Pakistan.
[2]Faculty of Intelligent Computing, Universiti Malaysia Perlis, 02600 Arau, Perlis, Malaysia.
[3]Department of Computer Science & IT, Baba Guru Nanak University, 39100 Nankana Sahib, Pakistan.
[4]Ecole Math´ematiques, Informatique, T´el´ecommunications de Toulouse, Universit´e de Toulouse, 31000 de Toulouse, France
*Correspondence: kalsoomsafdar11@gmail.com, usman1644@gmail.com

Citation | Imran. S, Safdar. K, Younus. M. U, "BISF-IoT: A Scalable Blockchain-Integrated Security Framework with Formal Guarantees", IJIST, Vol. 8 Issue 5 pp 1966-1994, August 2026
DOI | https://doi.org/10.33411/IJIST/1994
Received | July 23, 2026 **Revised |** Aug 09, 2026 **Accepted |** Aug 09, 2026 **Published |** Aug 16, 2026.

As IoT deployments rapidly expand, ensuring comprehensive end-to-end security across identity, authorization, communication, integrity, and auditability is critical. This paper presents BISF-IoT, a Blockchain-Integrated Security Framework that utilizes a permissioned ledger as a tamper-evident control plane while keeping high-volume telemetry and raw logs off-chain. BISF-IoT integrates decentralized identity (DID) management, capability-based authorization with explicit revocation under a freshness bound Δ, and secure-channel identity binding for MQTT/CoAP edge devices. Formal game-based proofs establish five core security properties: DID authenticity, authorization soundness, revocation safety, tamper-evident logging, and auditable anomaly alert non-repudiation. Performance evaluation through simulation with up to 10,000 devices demonstrates near linear scalability, processing up to 160,000 transactions per day with a scaling efficiency of ~0.90–1.00. Authorization latency increases moderately from 120 ms at 100 devices to 650 ms at 10,000 devices. Therefore, it remains within practical operational limits. Blockchain storage grows steadily at approximately 37–42 MB/day by storing compact Merkle commitments and security artifacts while avoiding raw data bloat. Meanwhile Log verification time exhibits sub-linear growth, with verification cost per entry decreasing from 0.200 ms to 0.055 ms as log size increases from 100 to 10,000 entries, reflecting efficient Merkle inclusion proof mechanisms. These results confirm BISF-IoT's capability to provide scalable, secure, and verifiable control-plane operations suitable for large-scale IoT environments.

Keywords: Blockchain, IoT, Decentralized, Capability, Authorization, Revocation, Tamper, Auditable

Introduction:

The Internet of Things (IoT) is rapidly expanding in domains such as industrial automation, smart healthcare, and connected infrastructure. Securing IoT deployments remains challenging because attacks can exploit weaknesses across the full-stack device identity lifecycle, authorization and access control, secure communication, integrity evidence for auditing, and the verifiability of incident evidence. In practice, strong protection at one layer can be undermined by weaknesses in adjacent layers: for example, an adversary may target authorization state, replay or spoof authenticated communications, or generate untrustworthy audit evidence when logs are not tamper-evident.

Recent studies increasingly investigate blockchain-based solutions for IoT security, particularly to replace centralized trust anchors with replicated and tamper-evident control state. Surveys and reviews highlight that blockchain–IoT designs often aim to improve integrity, access control, and auditability while addressing security and trust concerns across heterogeneous IoT environments [1][2], [3][4]. However, many existing works focus on a single security dimension (e.g., identity, access control, or logging) rather than specifying end-to-end integration rules that connect these dimensions under a unified verification process. For instance, decentralized identity and secure data storage are studied using blockchain-based models [5], while access control architectures are developed for IoT-healthcare [6], secure access control management [7], and fine-grained knowledge sharing in cloud edge orchestrated settings [8]. Similarly, revocation mechanisms and privacy-enhancing access control are investigated to improve authorization lifecycle handling [9][10][11].

Beyond access control state, secure communication assumptions are essential. Lightweight blockchain-assisted authentication and authorization mechanisms designed for MQTT security have been proposed to reduce attack impact in IoT messaging pipelines [12]. Related work also studies blockchain-based mitigation frameworks against deauthentication attacks and other message-centric disruptions [13]. For OTA and message integrity, MQTT-integrated Merkle-based approaches have been proposed to strengthen verifiability of firmware/security updates [14]. Nevertheless, even when these components are strong individually, integrated security guarantees remain unclear if the framework does not explicitly bind communication identity to authorization identity and to the evidence used during later auditing.

Another recurring theme is the need for tamper-evident integrity evidence. Several works use Merkle-tree-related mechanisms to support integrity verification and tamper resistance in IoT logging and update flows [14][7], while privacy-oriented designs employ zero-knowledge techniques to reduce link ability in authorization workflows [11]. Finally, formal security analysis is still not universal across the literature, and when formal methods are present they are often limited to specific protocols or modules rather than a complete cross-layer framework [4].

Despite extensive literature on blockchain-assisted IoT security, many proposed systems lack an end-to-end integration of core security subsystems. Identity management, secure communication, integrity logging, and anomaly detection are often treated as separate modules, and security claims are frequently stated informally.

As a result, it is unclear whether an integrated deployment would satisfy cross-layer guarantees such as: (i) only authorized and authenticated subjects can successfully actuate devices, (ii) revocations reliably prevent continued access after ledger finality, (iii) device logs are tamper-evident and auditable, and (iv) anomaly alerts can be verified and attributed to specific model versions and committed inputs.

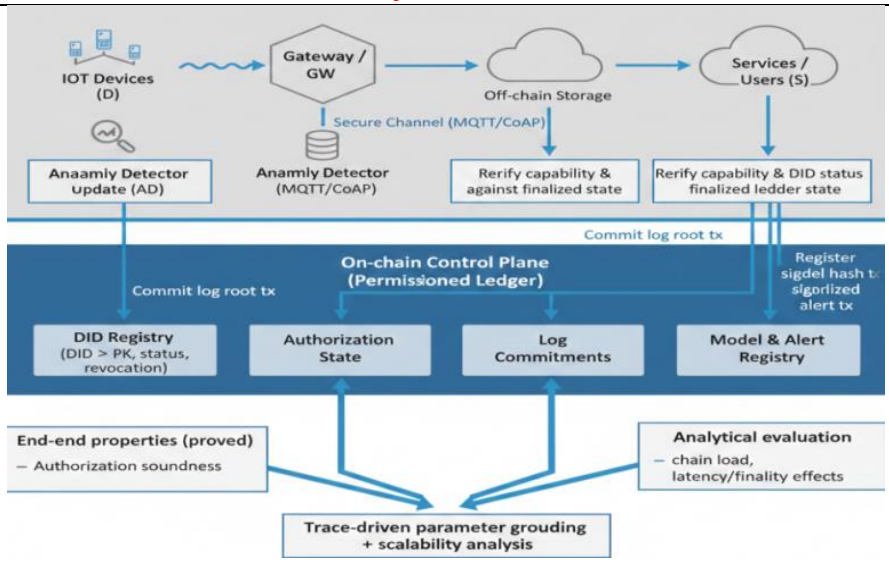

Figure Error! No text of specified style in document.**.** Off-Chain Data Plane

This research addresses the integration gap by designing a single framework that links these subsystems through a common trust root, decentralized identity anchored on a ledger and by proving end-to-end properties under explicit assumptions.

Industry demand for comprehensive IoT security guidance emphasizes baseline capabilities. This includes device identification, secure communications, logging, and update/maintenance processes as a system, not as independent features. This motivates architectures that unify identity, policy enforcement, and audit evidence rather than deploying isolated controls [NISTIR 8259A, 2020].

Cross-layer security enables novel guarantees as shown in Figure 1. Integration is not just "more components." It enables new end-to-end claims that inherently span multiple layers. For example, a device accepting an actuation command should imply the requester was authenticated to a registered identity. Meanwhile the request was authorized by finalized policy state, and the event is audit-evidenced. Such properties naturally lead to formal definitions and proofs (game-based or composable models),which are often missing from framework-oriented proposals [Canetti, 2001].

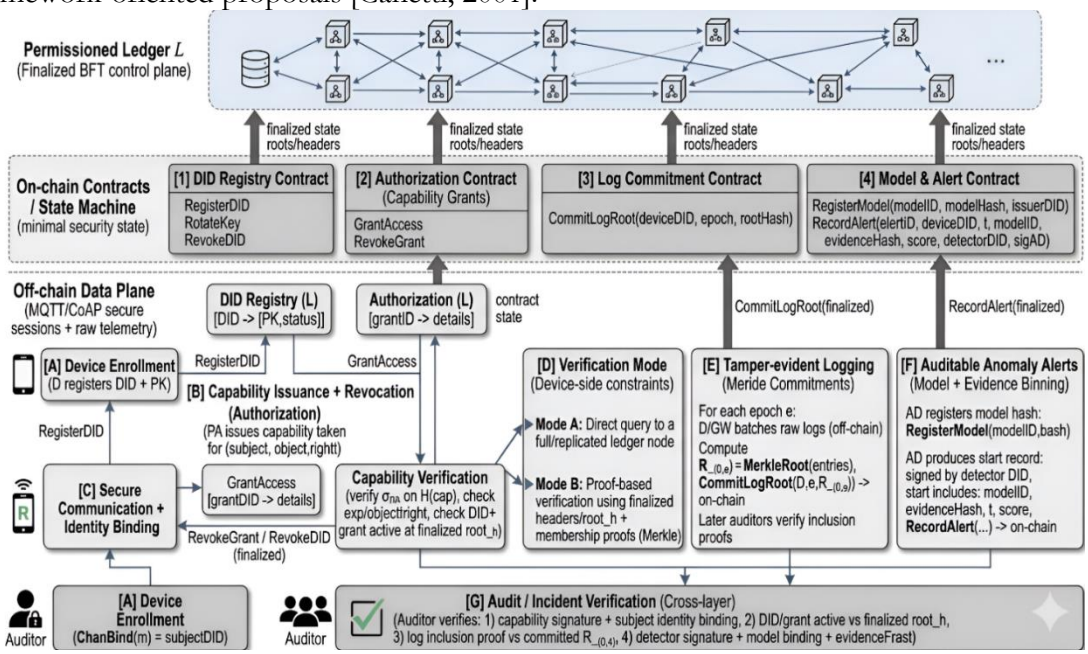

Figure 2. BISF-IoT Detailed Framework Diagram (Conceptual Control Plane + Data Plane)

On chain "security control plane" (minimal state): The ledger L stores only what is required to make verification possible and tamper-evident: DID Registry contract: RegisterDID, RotateKey, RevokeDID enables DID authenticity + revocation safety. Authorization contract: GrantAccess, RevokeGrant enables capability correctness + revocation safety. Log Commitment contract: CommitLogRoot(deviceDID, epoch, rootHash) enables tamper-evident logging without storing raw logs. Model & Alert contract: RegisterModel(...), RecordAlert(...) enables auditable anomaly alerts bound to model versions and evidence hashes as shown in Figure 2.

Off-chain data-plane flows (what happens during operation) Enrollment (DID registration): device D registers its DID and public key on-chain. Capability issuance: Policy Authority (PA) issues signed capability tokens (bound to grant semantics: subjectDID, objectDID/right, exp, etc.). Secure communication & binding: the request is sent over MQTT/CoAP secure channels (DTLS/OSCORE abstraction).

The device enforces: $cap.subjectDID = ChanBind(m)$ Device authorization verification: the device checks the capability signature and then checks DID/grant status against a finalized ledger root. Revocation safety with freshness bound Δ: if revocation happened at height h_{rev}, devices accept only when they verify using finalized roots with $h \geq h_{rev} + \Delta$. Logging (Merkle commitments): raw logs stay off-chain; only Merkle roots are committed on-chain per epoch. Anomaly alerts: AD registers model hashes and later records signed alerts referencing modelID and evidenceHash. Auditors can verify authenticity and binding.

Verification modes (important for constrained devices) The framework includes two modes: Mode A (direct query): device/gateway queries ledger state for DID/grant/log roots. Mode B (proof-based): device uses finalized root h and membership proofs (Merkle-style) to verify state locally.

Cross-layer assurance (why the "integration" matters) My "acceptance ⇒ evidence chain" is strong: If the device accepts an action, then (under the proposed model) it implies: Identity authenticity via DID registry correctness, Authorization soundness via capability signature + ledger-consistent grant state, Secure-channel identity binding via ChanBind(m), Revocation-respecting freshness via Δ,an audit evidence availability via Merkle committed logs and auditable alerts.

This paper aims to address the persistent integration gap in blockchain-based IoT security by developing a comprehensive framework that unifies core security subsystems under a single control plane. The specific objectives are:

To design BISF-IoT, a scalable blockchain-integrated security framework that ensures end-to-end security guarantees spanning decentralized identity management, capability-based authorization with explicit revocation, secure communication identity binding, tamper-evident logging, and auditable anomaly detection.

To formally define and prove five core security properties DID authenticity, authorization soundness, revocation safety under a staleness bound Δ, tamper-evident logging, and auditable anomaly alert non-repudiation using game-based security models.

To implement an efficient on-chain control plane using a permissioned ledger that stores minimal security-critical state. While keeping high-volume telemetry and raw logs off-chain, thereby optimizing scalability and storage overhead.

To evaluate the performance of BISF-IoT through simulation with varying network sizes (100 to 10,000 devices), focusing on quantitative metrics including scalability (transactions per day), authorization latency, blockchain storage growth, and log verification efficiency.

To demonstrate that BISF-IoT achieves near-linear scalability, practical latency bounds, controlled storage overhead, and sub-linear log verification costs, validating its suitability for large-scale, security-sensitive IoT deployments.

These objectives collectively aim to provide a unified, formally verified, and practically scalable security framework tailored to the complex requirements of modern IoT ecosystems.

BISF-IoT distinguishes itself from existing blockchain-IoT security frameworks through its comprehensive end-to-end integration of core security subsystems under a unified control plane. Unlike many prior works that address isolated aspects such as identity management, access control, or tamper-evident logging independently, BISF-IoT explicitly links decentralized identity (DID) management, capability-based authorization with explicit revocation under a freshness bound Δ, secure communication identity binding, tamper-evident logging, and auditable anomaly detection within a single permissioned ledger framework. This integration enables formal game-based proofs of five critical security properties DID authenticity, authorization soundness, revocation safety, tamper-evident logging, and anomaly alert non-repudiation, providing rigorous end-to-end security guarantees rarely achieved in prior research.

Furthermore, BISF-IoT optimizes scalability and storage by maintaining a minimal on-chain control plane that stores only security critical state and compact Merkle commitments, while offloading high volume telemetry and raw logs off chain. Performance evaluations demonstrate near-linear scalability to 10,000 devices, practical authorization latency, and controlled blockchain storage growth. Meanwhile the efficiency of sub-linear log verification, validating its suitability for large-scale, security-sensitive IoT deployments.

In addition, the BISF-IoT's novelty lies in its formally verified, scalable, and integrated blockchain-based security framework. This work holistically addresses identity, authorization, communication binding, integrity evidence, and auditability, thereby bridging the persistent integration gap evident in existing blockchain-IoT security solutions.

Related Work:

Blockchain-based IoT security research can be grouped into: (i) identity and access control, (ii) revocation and lifecycle enforcement, (iii) secure communication (including MQTT security), (iv) tamper evident logging and integrity evidence (often Merkle-based), (v) privacy-enhancing authorization (including ZK), and (vi) formal security analysis.

Identity and Access Control Using Blockchain-based identity management and access control have been studied as a way to reduce reliance on centralized identity stores. A systematic review on blockchain base decentralized identity models for secure IoT highlights. The role of ledger-anchored identity bindings and security trust improvements [5]. This Architecture focused on study and propose access control schemes for IoT environments, including healthcare-specific designs [6] and secure access control management mechanisms [7]. Beyond general access control, encryption-enabled knowledge-sharing for cloud edge orchestrated. Meanwhile IoT using blockchain has been explored to support privacy-preserving federated learning and access mediation [8]. Reviews focusing on broader blockchain IoT security and integration strategies furthermore it confirms that access control centric blockchain designs are common but vary widely in how they connect the identity, authorization, and verifiable evidence [1][2][3][4].

Revocation and Authorization Lifecycle Enforcement Revocation is a critical authorization lifecycle requirement in IoT deployments. Therefore, long lived authorization tokens increase the risk factor after policy change. Revocation capable authorization designs for IIoT data sharing emphasize fair and efficient revocable access control based on blockchain [9]. Permissioned revocation mechanisms designed to support access control in blockchain-based IoT systems have also been studied [15]. These works provide important

foundations for revocation but may not always connect revocation enforcement with secure communication identity binding and with tamper evidence used for audits.

MQTT Security and Secure Message Integrity Because many IoT deployments rely on message queues and publish/subscribe interactions, secure communication is often implemented at the MQTT layer. Recent work proposes lightweight two-way authorization and authentication mechanisms to enhance MQTT security for IoT systems [12]. Deauthentication-focused mitigation mechanisms based on blockchain have been studied to address attacks that disrupt session/service availability and session integrity [13]. In addition, Merkle-tree-based verifiability for MQTT-based secure OTA protocols has been proposed to improve integrity of update content and verification workflows [14]. These communication-layer advances are important, but end-to-end security requires that the authenticated sender identity is consistent with authorization identity and the evidence committed for audit.

Tamper-Evident Logging and Integrity Evidence is essential for incident response, compliance, and exploring why a device accepted an action. Prior work proposes tamper-proof logging using blockchain and Merkle-tree verification in IoT systems [7]. MQTT-integrated Merkle Tree approaches also contribute to verifiable integrity evidence for update workflows [14]. Such mechanisms support integrity, but integrated frameworks must also define how log commitments relate to authorization, acceptance and revocation respecting evidence.

Privacy that Preserves the Authorization and ZK Privacy preserving access control is increasingly studied to reduce link ability in IoT authorization workflows. A zero-knowledge based approach for enhancing account information anonymity in blockchain-based IoT access control demonstrates how privacy can be added to authorization systems [11]. Meanwhile Privacy enhanced role-based access control for IoT devices. Furthermore, practical authorization designs that incorporate privacy requirements [10]. However, privacy mechanisms must still interoperate with auditable evidence requirements; otherwise, incident verification may be difficult.

Formal Security Analysis Formal methods help provide stronger assurances about security properties of blockchain-based authorization protocols. A dedicated study on formal security analysis of blockchain-enabled access control protocols for IoT illustrates the value of rigorous definitions and analyses [4]. Still, integrated end-to-end security across identity, authorization, revocation, communication binding, and evidence verification often remains under specified in modular designs. Which motivates the BISF-IoT framework's unified modeling and proof goals. The comparison of blockchain–IoT security approaches with respect to BISF-IoT is given in Table 1.

Table 1. Comparison of blockchain–IoT security approaches with respect to BISF-IoT themes

Work/ Theme	Identity & Auth	Authorization & Revocation	MQTT/Secure Comms	Merkle Logging/Integrity Evidence	Privacy (ZK/Privacy)	Formal Security
Blockchain-IoT secure storage & access control (systematic) [5]	Decentralized identity focus	Access control emphasis (fine-grained)	Not primary	Not primary	Not primary	Not primary
Security review for blockchain-based IoT [1]	Broad identity/auth coverage	Broad access control themes	Mentions integration security needs	Broad audit/integrity discussion	Broad privacy discussion	Not specific
Blockchain SLR for IoT trust/security [2]	Identity and trust themes	Access-control-focused trends	Integration-oriented view	Integrity evidence discussed at high level	Privacy discussed at survey level	Not protocol-level
IoT-healthcare access control architecture [6]	Auth via blockchain mechanisms	Access control for healthcare	Not primary	Not primary	Not primary	Not primary
BloAC secure access control management [7]	Access control with blockchain	Access management lifecycle	Not primary	Tamper-proof logging + Merkle verification (as titled)	Not primary	Not primary
Secure MQTT + lightweight authorization/auth [12]	Auth for MQTT	Authorization at MQTT access layer	Primary: MQTT security	Potential integrity verification (not full framework)	Not primary	Not primary
Blockchain mitigation for deauthentication attacks [13]	Session/authentication context	Mitigation enabling authorization continuity	Primary: deauthentication attack model	Not primary	Not primary	Not primary
MQTT + MerkleTree secure OTA protocol [14]	Auth for MQTT OTA workflows	Authorization embedded in OTA flow	Primary: MQTT	MerkleTree-based verifiability for OTA	Not primary	Not primary
Access control revocation for	Authorization lifecycle	Revocation explicitly	Not primary	Not primary	Not primary	Not primary

IIoTdata sharing [9]		targeted				
Blockchain-enabled revocation/access control in IIoT contexts [15]	Authorization lifecycle	Efficient revocation mechanisms	Not primary	Not primary	Not primary	Not primary
Privacy-enhanced RBAC for IoT [10]	Identity mapped to roles	Role-based authorization	Not primary	Not primary	Privacy-enhanced access control	Not primary
Anonymity via zero-knowledge proofs [11]	Privacy of account/identity	Access control with privacy	Not primary	Not primary	ZK-driven privacy	Not primary
Formal security analysis of blockchain-enabled access control protocols [4]	Auth protocol analysis	Authorization correctness (formal)	Not primary	Not primary	Not primary	Primary: formal security analysis
Trust-aware explainable access control (IoMT) [16]	Trust-based identity/access decisions	Access control with trust/explainability	Not primary	Not primary	Not primary	Not primary

Materials and methods:

System Design Overview Within the operational framework, BISF-IoT establishes a division between different functional areas:

The on-chain control plane (permissioned ledger): identity enrolment list, authorization allowances and removals, Merkle log roots, and model/alert audit records.

Concerning the off-chain data plane: MQTT/CoAP secure sessions, raw telemetry, and raw log elements. The framework records only security-critical artifacts with low transaction rates on-chain. High-volume telemetry and raw logs remain off-chain even though high-volume telemetry and raw logs become capable of being authenticated through Merkle inclusion proofs. The entity symbol and its role in BISF-IoT are given in Table 2.

Table 2. Entities and roles in BISF-IoT

Symbol	Entity	Role	Trust notes
D	IoT Device	Enforces acceptance, verifies capabilities/proofs, optionally commits log roots	May be corrupted
S	Subject/Service	Requests access/actuation, presents capabilities	May be corrupted
PA	Policy Authority	Issues grants, revokes DID/grants, signs capability tokens	May be corrupted
AD	Anomaly Detector	Registers model hashes and records signed anomaly alerts	May be corrupted
GW (optional)	Gateway/Edge	Batches proofs/roots/log commitments	Not assumed trusted
L	Permissioned Ledger	Stores finalized state for DID, grants/revocations, commitments, alerts	BFT finality assumed

Cryptographic Building Blocks and Assumptions:

Assumptions of BISF-IoT include:

The digital signature scheme offers satisfactory EUF-CMA security.

The commitment hash function is collision-resistant.

The permissioned ledger ensures that finalization guarantees consistency of state (revocation and commitments cannot be undone after finalization).

On-Chain State Model (Minimal Control Plane)**component, state mapping and purpose is given in Table 3.**

Table 3. On-chain state components

Component	State mapping (conceptual)	Purpose
DID Registry	DID → (PK, status)	Identity authenticity and DID revocation
Authorization State	grantID→ (issuerDID, subjectDID, objectDID, right, exp, status)	Capability lifecycle and revocation
Log Commitments	(deviceDID, epoch) →MerkleRoot	Tamper-evident logging root
Model Registry	modelID→ (modelHash, status)	Model version integrity for alerts
Alert Records	alertID→ signed alert fields	Non-repudiation and auditability

Secure-Channel Identity Binding:

ChanBind(m) represents the DID that is authenticated via the secure channel used for request/message m. The BISF-IoT scheme ensures:

$$cap.subjectDID = ChanBind(m)$$

Equation 1:

The communication identity is bound to the authorization subject identity whose authorization status has been validated using the ledger.

Capability-Based Authorization with Revocation:

The Policy Authority signs the capability token because the capability token connects authorization meanings together. The capability token must fulfill specific conditions before the device accepts it. Legitimacy of the signature on the capability token is demanded by the system while the subject identity binding, which involves secure channel binding, is also checked. The capability token remains useful only when the DID and the grant show a functioning condition against a finalized ledger root. It has been observed that freshness and expiry legitimacy are checked by the machine at the same time. Safety regarding revocation is enforced by the system through a finalized height because the system must respect a staleness bound Δ.

Tamper-Evident Logging via Merkle Commitments: For each e, device D batches log entries and commits:

$$R_{(D,e)} = MerkleRoot\left(Entries_{(D,e)}\right)$$

Equation 2

Auditors verify an individual log entry using Merkle inclusion proofs against the finalizedon-chain root.

Auditable Anomaly Alerts:

The anomaly detector AD uploads model hashes in blockchain. Alerts are signed and contain:

DID of the anomaly detector,

ModelID (corresponding to the registered modelHash),

EvidenceHash (associated with the committed evidence/log epoch window),

Timestamp & Scores.

Security Evaluation Method (Game-Based):

The BISF-IoT framework provides for five inter-related characteristics in game-theoretic fashion:

DID impersonation resistance

End-to-end authorization security

Revocation safety with Δ freshness guarantee

Merkle log tamper-resistance

Anomaly auditability without model repudiation

Performance Evaluation (Simulation):

The simultation evaluates IoT device populations ranging from 100 to 10,000 devices. This simulation evaluates scalability in terms of transactions per day because counting the daily activity matters. Authorization latency measured in ms is calculated during the process by simulation. Measurement of blockchain storage growth in MB happens over time while log verification time in ms is compared against log size.

Here is the complete workflow of the proposed BISF-IoT framework, detailing the interaction among all modules:

System Architecture Overview:

BISF-IoT divides its architecture into two distinct planes: the on-chain control plane and the off-chain data plane. The control plane is realized via a permissioned ledger that stores minimal security-critical state, while the data plane handles high-volume telemetry and raw logs off-chain, ensuring scalability and efficient storage.

On-Chain Control Plane Components:

DID Registry Contract: Manages decentralized identity (DID) registration, key rotation, and revocation, ensuring identity authenticity and revocation safety.

Authorization Contract: Handles capability-based authorization grants and revocations, maintaining lifecycle correctness and enabling explicit revocation under a freshness bound Δ.

Log Commitment Contract: Commits Merkle roots of batched log entries per device and epoch, enabling tamper-evident logging without storing raw logs on-chain.

Model and Alert Contract: Registers anomaly detection model hashes and records signed anomaly alerts linked to specific model versions and evidence hashes, supporting auditable anomaly alert non-repudiation.

Off-Chain Data Plane Components:

IoT Devices: Enforce authorization by verifying capability tokens against finalized ledger states and validate secure-channel identity binding (ChanBind(m)) for MQTT/CoAP edge devices. Devices batch log entries off-chain and commit Merkle roots on-chain.

Policy Authority (PA): Issues and revokes signed capability tokens bound to subject and object DIDs with defined rights and expiration, ensuring authorization soundness.

Anomaly Detector (AD): Uploads model hashes and records signed anomaly alerts referencing model and evidence hashes for auditability.

Gateways/Edge Nodes (Optional): Batch proofs, roots, and log commitments to optimize communication with the ledger, though not assumed fully trusted.

Workflow and Interactions:

Enrollment: Devices register their DIDs and associated public keys on-chain via the DID Registry contract.

Capability Issuance: The Policy Authority issues signed capability tokens defining access rights, which devices verify against the ledger's finalized state before accepting actions.

Secure Communication and Identity Binding: All requests are sent over secure MQTT/CoAP channels (e.g., DTLS/OSCORE), with the communication identity cryptographically bound to the authorization identity (ChanBind(m)).

Authorization Verification: Devices check capability token signatures and ledger state for active grants, ensuring revocation safety by verifying against finalized ledger roots with freshness bound Δ.

Logging: Devices batch raw logs off-chain and periodically commit Merkle root hashes on-chain, enabling later verification via Merkle inclusion proofs without storing bulky raw data on the ledger.

Anomaly Alerting: The anomaly detector registers model hashes and records signed alerts on-chain, linking alerts to specific model versions and evidence to support auditability.

Verification Modes: Devices or gateways may verify authorization and log states either by directly querying the ledger (Mode A) or by locally verifying finalized root hashes and membership proofs (Mode B), accommodating constrained environments.

Security and Trust Assumptions:

The framework assumes standard cryptographic security (EUF-CMA for signatures, collision resistance for hash functions) and permissioned ledger finality guarantees, ensuring consistency and immutability of on-chain state. Devices and entities may be corrupted, but the ledger and cryptographic proofs provide tamper-evident and verifiable control.

This structured workflow integrates identity management, authorization, binding secure communication, tamper-evident logging, and auditable anomaly detection into a unified, scalable framework suitable for large IoT deployments.

The simulation setup description should be expanded to include the following implementation details to ensure reproducibility and clarity:

Blockchain Platform: Idealized Permissioned Ledger A permissioned ledger modeled as an abstract state machine with:

finalized blocks/roots at ledger height h,

read access to finalized state only,

deterministic finality (no reorgs under the assumed consensus correctness).

(No concrete platform such as Hyperledger Fabric / Ethereum was executed.)

Consensus Mechanism: PBFT-style BFT with Deterministic Finality (Permissioned BFT): The ledger is assumed to follow a **BFT protocol conceptually equivalent to PBFT**, where once a block/state is *finalized*, it is not reverted.
Finality time is modeled as T_f and used in latency:

$$T_{\text{policy}} \approx T_f + T_p$$

Simulation Software: A custom simulation/parameter-evaluation tool was used to model:
transaction generation rates (e.g., $nE + u + a$),
authorization latency impact from finality,
log commitment verification cost (Merkle inclusion proof growth).
Hardware Configuration:
(CPU)Intel Xen Processor
(Ram) 12 GB Ram
(Storage)256 GB SSD
Parameter Settings: List key parameters is given in Table 4.

Table 4. Simulation Parameter Settings

Parameter	Symbol	Value / Range	Description & Justification
Network Size	n	100 – 10,000	Number of IoT devices, scaled to observe system behavior under large deployments.
Log Epochs/day	E	12 (every 2 hours)	Frequency of Merkle root commitments per device per day. Balances audit granularity and on-chain load.
Policy Updates/day	u	$0.1 \times n$	Dynamic grant issuances/revocations (10% of network daily), reflecting moderate policy churn.
Anomaly Alerts/day	a	$0.05 \times n$	Alert rate derived from trace-driven grounding (e.g., CIC-IDS2017 attack ratios applied to IoT scale).
Finality Time	T_f	2 seconds	Deterministic finality time for a PBFT-style permissioned ledger (typical enterprise BFT benchmark).
Proof Propagation	T_p	1 second	Network delay to distribute finalized headers/Merkle proofs to constrained devices or gateways.
Staleness Bound	Δ	3 blocks	Max staleness window for revocation safety (e.g., 3 finality blocks = 6 seconds delay tolerance).
Sig. Verify Time	$T_{sig-verify}$	1.5 ms	Time to verify an ECDSA/Ed25519 signature on a constrained IoT device (e.g., ARM Cortex-M series).
Hash Time	T_{hash}	0.001 ms	Time to compute a SHA-256 hash on constrained hardware.
Merkle Tree Size	N	$10^3 – 10^5$	Number of leaves (e.g., active grants or log entries) in the authenticated data structure.
Hash Output Size	$	H	$
Capability Token	$	cap	$

Size			

Assumptions and Proof Model:

Let λ be the security parameter. We adopt the standard cryptographic and system assumptions for proofing.

Assumption 1 (EUF-CMA signatures). The signature scheme $\Sigma = (\text{KeyGen},\text{Sign},\text{Verify})$ is existentially unforgeable under chosen-message attacks. For any probabilistic polynomial-time (PPT) adversary $\mathcal{A}$, there exists a negligible function $\text{negl}_1(\lambda)$ such that

$$Adv\Sigma EUF - CMA(A) = \Pr\left[Verify\big(PK, \sigma, H(m)\big) = 1 \wedge (m,\sigma) \notin \frac{}{Q_{\backslash textSign}}\right] \leq negl_1(\lambda)$$

Equation 3

where Q_{Sign} denotes the set of messages–signature pairs obtained from the signing oracle.

Assumption 2 (Collision-resistant hashing and Merkle soundness). The hash function $H:\{0,1\}^* \to \{0,1\}^k$ is collision-resistant. For any PPT adversary $\mathcal{A}$, $\text{Adv}_H^{\text{CR}}(\mathcal{A}) \leq \text{negl}_2(\lambda)$. Moreover, for any committed Merkle root R over a set of leaves L, it is infeasible to produce (x, π) such that $\text{MerkleVerify}(R, x, \pi) = 1$ and $x \notin L$, except with probability bounded by $\text{negl}_2(\lambda)$.

Assumption 3 (Ledger finality and state consistency). The permissioned ledger $\mathcal{L}$ maintains a finalized state root root_h at height h. Once root_h is finalized, the state mapped by $\text{StateAt}(\text{root}_h, \cdot)$ does not revert, and no PPT adversary can convince a verifier to accept two contradictory finalized values for the same key under the same root_h without violating A2 or the ledger's consensus correctness. We model this as an abstract predicate $\text{LedgerConsistent}(\text{root}_h)$; any deviation implies breaking A2 or the ledger fault assumption.

Assumption 4 (Secure-channel identity binding). The secure communication layer (DTLS/OSCORE) guarantees that for any message m, the verified sender identity $\text{ChanBind}(m)$ equals the DID that authenticated the channel, unless the adversary possesses the corresponding long-term credentials or breaks the channel's authentication guarantees. Formally, no PPT adversary can cause an honest receiver to accept $\text{ChanBind}(m) = DID^*$ for an uncorrupted DID^* without forging the channel credentials.

Devices verify authorization state either by (i) direct query to a trusted/full node returning $\text{StateAt}(\text{root}_h, \cdot)$, or (ii) proof-based verification using a finalized header root_h and Merkle/membership proofs. In both cases, acceptance of an incorrect finalized value implies breaking A2 or A3.

Theorem 1 DID Impersonation Resistance:

Game 1 (EXP^{DID}). Challenger $\mathcal{C}$ initializes $\mathcal{L}$, registers a set of active DIDs $\{DID_x\}$ with public keys PK_x, and provides $\mathcal{A}$ with oracles $Corrupt(\cdot)$, $LedgerRead(\cdot)$, and optionally a signing oracle for uncorrupted identities. $\mathcal{A}$ outputs (DID^*, m^*, σ^*). $\mathcal{A}$ wins if:

DID^* is registered, active, and not corrupted,

$\text{Verify}\big(PK_{DID^*}, \sigma^*, H(m^*)\big) = 1$, and

(m^*, σ^*) was not obtained from a legitimate signing oracle for DID^*

$$\text{Let } \text{Adv}^{\text{DID}}(\mathcal{A}) = \Pr\left[\text{Exp}_A^{\text{DID}}(\lambda) = 1\right]$$

Theorem 1. Under A1 and A3, for any PPT adversary $\mathcal{A}$,

$$\text{Adv}^{\text{DID}}(\mathcal{A}) \leq \text{Adv}_{\Sigma}^{\text{EUF-CMA}}(\mathcal{B}_1) + \Pr[\neg\text{LedgerConsistent}(\text{root}_h)] + \text{negl}(\lambda),$$

Equation 4

and hence $\text{Adv}^{\text{DID}}(\mathcal{A}) \leq \text{negl}(\lambda)$.

Proof. Suppose $\mathcal{A}$ wins Game 1 with non-negligible probability. Let root_h be the finalized root used by the verifier during the winning transcript. By A3, the public key PK_{DID^*} used by the verifier must equal the ledger's finalized DID registry value at root_h, unless $\neg\text{LedgerConsistent}(\text{root}_h)$ holds.

We distinguish two exhaustive cases:

Case 1: $\text{LedgerConsistent}(\text{root}_h)$ *holds.* Then PK_{DID^*} is the correct, ledger-finalized public key for DID^*. Since DID^* is uncorrupted, $\mathcal{A}$ does not possess sk_{DID^*}. The winning condition $\text{Verify}(PK_{DID^*}, \sigma^*, H(m^*)) = 1$ with (m^*, σ^*) not obtained from a signing oracle constitutes a valid EUF-CMA forgery against Σ. We construct $\mathcal{B}_1$ as follows:

$\mathcal{B}_1$ receives PK^* from the EUF-CMA challenger and sets $PK_{DID^*} = PK^*$.

$\mathcal{B}_1$ simulates the ledger, oracles, and registration state for $\mathcal{A}$, forwarding any signing queries for DID^* to its own EUF-CMA signing oracle.

When $\mathcal{A}$ outputs (DID^*, m^*, σ^*), $\mathcal{B}_1$ checks the winning conditions. If they hold and (m^*, σ^*) is fresh, $\mathcal{B}_1$ outputs (m^*, σ^*) as its forgery.

By construction, $\Pr[\mathcal{B}_1 \text{ wins}] \geq \Pr[\mathcal{A} \text{ wins} \wedge \text{LedgerConsistent}(\text{root}_h)]$.

Hence

$$\Pr[A\ wins \wedge LedgerConsistent(rooth)] \leq Adv_{\Sigma}^{EUF-CMA}(B1) + negl(\lambda) \quad \text{Equation 5}$$

Case 2: $\neg\text{LedgerConsistent}(\text{root}_h)$. This event occurs with probability bounded by the ledger's consistency violation probability, which is assumed negligible under the permissioned BFT fault model (or otherwise reduces to breaking A2/A3). Denote this probability by $\text{Adv}^{\text{LedgerIncons}}$.

Combining both cases via the law of total probability:

$$\text{Adv}^{\text{DID}}(\mathcal{A}) \leq \text{Adv}_{\Sigma}^{\text{EUF-CMA}}(\mathcal{B}_1) + \text{Adv}^{\text{LedgerIncons}} + \text{negl}(\lambda) \leq \text{negl}(\lambda). \quad \text{Equation 6}$$

Theorem 2 End-to-End Authorization Soundness:

Game 2 (EXP^{AuthZ}). $\mathcal{C}$ sets up honest entities and ledger state. $\mathcal{A}$ may corrupt parties, submit ledger transactions (GrantAccess, RevokeGrant, RevokeDID, etc.), and schedule message delivery. $\mathcal{A}$ outputs a transcript τ causing an honest device D to process a request m with capability cap and PA signature σ_{PA}, verified against finalized root root_h. $\mathcal{A}$ wins if D accepts, yet at least one acceptance condition is false in the finalized state referenced by root_h.

The device acceptance predicate $\text{Acc}(D, m, \text{cap}, \sigma_{PA}, \text{root}_h) = 1$ holds iff:

(Sig) $\text{Verify}(PK_{PA}, \sigma_{PA}, H(\text{cap})) = 1$,

(Bind) $\text{cap.subjectDID} = \text{ChanBind}(m)$,

(Object) $\text{cap.objectDID} = DID_D$,

(Active) subject DID and issuer DID are active at root_h,

(Grant) the grant referenced by cap.grantID is active at root_h, matches (subjectDID, objectDID, right), and

(Exp) cap.exp $\geq$ now.

$$\text{Let } \text{Adv}^{\text{AuthZ}}(\mathcal{A}) = \Pr\left[\text{Exp}_A^{\text{AuthZ}}(\lambda) = 1\right] \quad \text{Equation 7}$$

Theorem 2. Under A1–A4, for any PPT adversary $\mathcal{A}$, $\text{Adv}^{\text{AuthZ}}(\mathcal{A}) \leq \text{negl}(\lambda)$.

Proof. Assume $\mathcal{A}$ wins Game 2. Then there exists a finalized root root_h and a request m such that D accepts while at least one condition in {Sig, Bind, Object, Active, Grant, Exp} is false at root_h. We perform a case analysis on the first failed condition that enables acceptance.

Case A: Sig *is false but device accepts.* If PK_{PA} is the ledger-finalized public key of an uncorrupted PA, acceptance implies $\text{Verify}(PK_{PA}, \sigma_{PA}, H(\text{cap})) = 1$ with $(\text{cap}, \sigma_{PA})$ not obtained from a legitimate PA signing oracle. This is a EUF-CMA forgery, contradicting A1. If instead the verifier used an incorrect PK_{PA} while believing it to be ledger-finalized, then $\neg\text{LedgerConsistent}(\text{root}_h)$ holds, contradicting A3 (and A2 for proof-based verification).

Case B: Active *or* Grant *is false but device accepts.* At root_h, the DID or grant is revoked/inactive/nonexistent. Acceptance implies the device was convinced of an active status. In proof-based mode, $\mathcal{A}$ must produce a membership/non-membership proof for an incorrect value, which contradicts Merkle soundness (A2) or ledger consistency (A3). In direct-query mode, acceptance implies the ledger returned an inconsistent finalized state, contradicting A3. Thus, this case reduces to breaking A2/A3.

Case C: Bind *is false but device accepts.* That is, $\text{cap.subjectDID} \neq \text{ChanBind}(m)$. Since the device enforces binding, acceptance implies $\text{ChanBind}(m)$ was forced to equal cap.subjectDID without valid channel credentials for that DID. This contradicts A4.

Case D: Object *or* Exp *mismatch but device accepts.* Acceptance despite a violated algorithm check contradicts the specified verification algorithm by construction. In the formal model, we treat the verifier code as correct; implementation bugs are out of scope for provable-security claims. Hence this case contributes zero probability to a successful cryptographic attack.

Since the cases are exhaustive and mutually exclusive with respect to the first violated condition that enables acceptance, we apply a union bound:

$$Adv^{AuthZ}(A) \leq Adv_{\Sigma}^{EUF-CMA} + \Pr[\neg LedgerConsistent] + \Pr[break\ A4] + negl(\lambda)$$

Equation 8

Each term is negligible under A1–A4, yielding $\text{Adv}^{\text{AuthZ}}(\mathcal{A}) \leq \text{negl}(\lambda)$.

Theorem 3 Revocation Safety with Staleness Bound Δ

Game 3 ($\text{Exp}^{\text{Revoke}\Delta}$). $\mathbf{C}$ initializes state as in Game 2. A revocation (DID or grant) is finalized at height h_{rev}. $\mathcal{A}$ outputs a transcript causing an honest device to accept a request using the revoked identity/grant while verifying against a finalized root root_h of height h. $\mathcal{A}$ wins if the device accepts and $h \geq h_{\text{rev}} + \Delta$.

Let $\text{Adv}^{\text{Revoke}\Delta}(\mathcal{A}) = \Pr\left[\text{Exp}_{A}^{\text{Revoke}\Delta}(\lambda) = 1\right]$.

Theorem 3. Under A1–A4 and assuming verifiers enforce freshness $h \geq h_{\text{rev}} + \Delta$, for any PPT adversary $\mathcal{A}$,

$$\text{Adv}^{\text{Revoke}\Delta}(\mathcal{A}) \leq \text{negl}(\lambda).$$

Proof. By contract semantics (Chapter 4), revocation is monotonic: once a DID or grant is marked revoked at finalized height h_{rev}, the state at every finalized height $h' \geq h_{\text{rev}}$ records status = revoked.

Assume $\mathcal{A}$ wins. Then there exists a finalized root root_h with $h \geq h_{\text{rev}} + \Delta$ such that the verifier, using root_h, accepts a request relying on a revoked DID/grant. At root_h, the status is revoked. For acceptance to occur, one of the following must hold:

The verifier is convinced that the revoked object is active. This requires either a forged membership proof for an incorrect active status (contradicting A2) or an inconsistent finalized state returned to the verifier (contradicting A3). In direct-query mode, this directly violates A3.

The verifier bypasses the status check due to a signature or binding forgery. A forged capability signature contradicts A1; a broken channel binding contradicts A4.

Formally, let E_{active} be the event that the verifier accepts a false active status at root_h. Then

$$\Pr[\mathcal{A}\text{ wins}] \le \Pr[E_{\text{active}} \wedge \text{LedgerConsistent}(\text{root}_h)] + \Pr[\neg\text{LedgerConsistent}(\text{root}_h)] + \Pr[\text{forgery}]$$

The first term is bounded by Adv_H^{CR} or Merkle soundness (A2), the second by ledger consistency (A3), and the third by A1/A4. All are negligible. Hence $\text{Adv}^{\text{Revoke}\Delta}(\mathcal{A}) \le \text{negl}(\lambda)$.

Theorem 4 Tamper-Evident Logging:

Game 4 (Exp^{Log}). For an honest device D and epoch e, $\mathcal{C}$ runs the honest logging procedure, producing $\text{Entries}_{D,e}$ and committing $R_{D,e}$ on-chain. $\mathcal{A}$ outputs (e^*, π^*). $\mathcal{A}$ wins

$$\text{if MerkleVerify}\left(R_{D,e}, e^*, \pi^*\right) = 1 \text{ and } \frac{e^*}{\text{Entries}_{D,e}}$$

Let $\text{Adv}^{\text{Log}}(\mathcal{A}) = \Pr\left[\text{Exp}_A^{\text{Log}}(\lambda) = 1\right]$

Theorem 4. Under A2 and A3, for any PPT adversary $\mathcal{A}$,

$$\text{Adv}^{\text{Log}}(\mathcal{A}) \le \text{Adv}_H^{\text{CR}}(\mathcal{B}) + \Pr\left[\neg\text{LedgerConsistent}\left(\text{root}_{h_{\text{commit}}}\right)\right] + \text{negl}(\lambda),$$

and thus $\text{Adv}^{\text{Log}}(\mathcal{A}) \le \text{negl}(\lambda)$.

Proof. Let $\text{root}_{h_{\text{commit}}}$ be the finalized root at which $R_{D,e}$ is committed. The commitment is immutable under A3: once finalized, $R_{D,e}$ cannot be altered without violating ledger consistency.

Assume $\mathcal{A}$ wins. Then (e^*, π^*) verifies under the committed $R_{D,e}$ despite $e^* \notin \text{Entries}_{D,e}$. We distinguish two exhaustive possibilities:

Root integrity holds: $R_{D,e}$ is exactly the Merkle root of $\text{Entries}_{D,e}$. Producing a valid inclusion proof for a non-member implies either a hash collision in the Merkle tree construction or a break in Merkle soundness. Under A2 (collision-resistant H), the probability of such an event is bounded by $\text{Adv}_H^{\text{CR}}(\mathcal{B})$ for some PPT reducer $\mathcal{B}$.

Root integrity is violated: The verifier is tricked into using a different root than the finalized $R_{D,e}$. This requires $\neg\text{LedgerConsistent}\left(\text{root}_{h_{\text{commit}}}\right)$, i.e., the ledger returned or accepted contradictory finalized state, contradicting A3.

By the law of total probability:

$$\text{Adv}^{\text{Log}}(\mathcal{A}) \le \text{Adv}_H^{\text{CR}}(\mathcal{B}) + \Pr\left[\neg\text{LedgerConsistent}\left(\text{root}_{h_{\text{commit}}}\right)\right] + \text{negl}(\lambda) \le \text{negl}(\lambda).$$

Equation 9

Theorem 5 Auditable Anomaly Alerts:

Game 5 ($\boldsymbol{Exp}^{\text{Alert}}$). $\mathcal{C}$ registers detector DIDs and model registry entries (modelID,modelHash $= H(\text{model})$,issuerDID). $\mathcal{A}$ may corrupt detectors and submit ledger transactions. $\mathcal{A}$ outputs an alert record (or transcript causing ledger acceptance):

$$\text{alert} = \langle \text{deviceDID}, t, \text{modelID}, \text{evidenceHash}, \text{score}, \text{detectorDID} \rangle, \sigma_{AD},$$

Equation 10

Such that the alert is accepted/recorded, yet at the verified finalized state at least one of the following holds:

$\text{Verify}\left(PK_{\text{detectorDID}}, \sigma_{AD}, H(\text{alert})\right) = 0,$

detectorDID is not active,

modelID is not active/registered, or

the model binding is inconsistent (e.g., modelID does not map to the claimed modelHash), or

the evidence-hash linkage rule is violated.

$$\text{Let Adv}^{\text{Alert}}(\mathcal{A}) = \Pr\left[\text{Exp}_A^{\text{Alert}}(\lambda) = 1\right]$$

Theorem 5. Under A1–A3, for any PPT adversary $\mathcal{A}$,

$$\mathrm{Adv}^{\mathrm{Alert}}(\mathcal{A}) \leq \mathrm{negl}(\lambda).$$

Proof. Acceptance of an alert on-chain requires:
detectorDID active and resolving to $PK_{\mathrm{detectorDID}}$ at the finalized root,
$\mathrm{Verify}\left(PK_{\mathrm{detectorDID}}, \sigma_{AD}, H(\mathrm{alert})\right) = 1$,
modelID active and resolving to a registered modelHash at the finalized root,
(Optionally) evidence-hash linkage to a committed epoch root satisfies the framework's format rule.

If $\mathcal{A}$ wins, at least one required condition is violated while the alert is still accepted. We analyze the possible **violations:**

Case 1: *Signature verification fails but alert accepted.* This implies σ_{AD} verifies under an uncorrupted detector's public key without that detector having signed alert, i.e., a EUF-CMA forgery, contradicting A1.

Case 2: *Registry inconsistency.* The verifier uses an incorrect $PK_{\mathrm{detectorDID}}$ or incorrect modelHash for a given modelID while believing the state is ledger-finalized. This contradicts A2/A3 (state proof soundness or ledger consistency). In proof-based mode, it requires a forged authenticated dictionary/Merkle proof; in direct-query mode, it requires inconsistent finalized state.

Case 3: *Model/evidence substitution that still verifies.* Any attempt to replace the model or evidence while preserving the same hash commitment implies finding a collision in H or breaking Merkle soundness for the evidence linkage structure, contradicting A2.

These cases are exhaustive for winning conditions. Applying a union bound:

$$\mathrm{Adv}^{\mathrm{Alert}}(\mathcal{A}) \leq \mathrm{Adv}_{\Sigma}^{\mathrm{EUF\text{-}CMA}} + \Pr[\neg \mathrm{LedgerConsistent}] + \mathrm{Adv}_{H}^{\mathrm{CR}} + \mathrm{negl}(\lambda) \leq \mathrm{negl}(\lambda).$$

Equation 11

Integrated End-to-End Corollary:

Corollary 1 (End-to-End Accountable Authorization): Under A1–A4, if an honest device D accepts a request, then there exists publicly verifiable evidence consisting of:
the capability token cap,
the issuer signature σ_{PA},
a finalized ledger root/proof showing issuer DID and grant status at acceptance time,

such that any third-party auditor can verify that the action was authorized under the binding identity, or attribute the issuance to the accountable PA if an improper grant was issued. Moreover, the acceptance implies the conjunction:

$$Acc \Rightarrow (ChanBind(m) = subjectDID) \wedge (valid\ \sigma PA) \wedge \left(\frac{grant}{DID\ active\ at\ rooth}\right) \wedge (revocation - respecting\ freshness) \wedge (auditable\ evidence)$$

Proof. By Theorem 2, acceptance implies signature validity, ledger-consistent status checks, and secure-channel identity binding. By Theorem 3, if a revocation occurred at h_{rev}, acceptance using a finalized root with $h \geq h_{\mathrm{rev}} + \Delta$ is infeasible unless A1–A4 are broken. By Theorem 4, any log commitment referenced for audit evidence is tamper-evident. By Theorem 5, any anomaly alert linked to the request is attributable to a specific detector, model version, and evidence window. Ledger finality (A3) preserves all referenced on-chain state and signatures, making the evidence publicly verifiable and non-repudiable. Hence, the cross-layer implication holds except with negligible probability and the proof structure summary is given in Table 5, where Table 6 consists on security properties and bad events.

Table 5. Proof Structure Summary

Experiment	Property	Broken assumption(s) if adversary wins
$\mathrm{Exp}^{\mathrm{DID}}$	DID impersonation resistance	A1 (EUF-CMA) or A3 (ledger

		consistency)
Exp^{AuthZ}	End-to-end authorization soundness	A1, A2/A3, or A4
$Exp^{Revoke\Delta}$	Revocation safety after $+\Delta$	A2/A3 (state consistency) + A1/A4 (token/channel binding)
Exp^{Log}	Tamper-evident logging	A2 (Merkle soundness/collision resistance) + A3 (root immutability)
Exp^{Alert}	Auditable anomaly alerts	A1 (detector signature) or A2/A3 (model/evidence binding)
Corollary 1	Cross-layer accountable authorization	A1–A4 jointly

Experimental Results:

Ends-to-End Acceptance and Audit Chain (Integration Effect):

End-to-End Acceptance and Audit Chain (device acceptance implies a chain of verifiable facts) structure is shown in Figure 3.

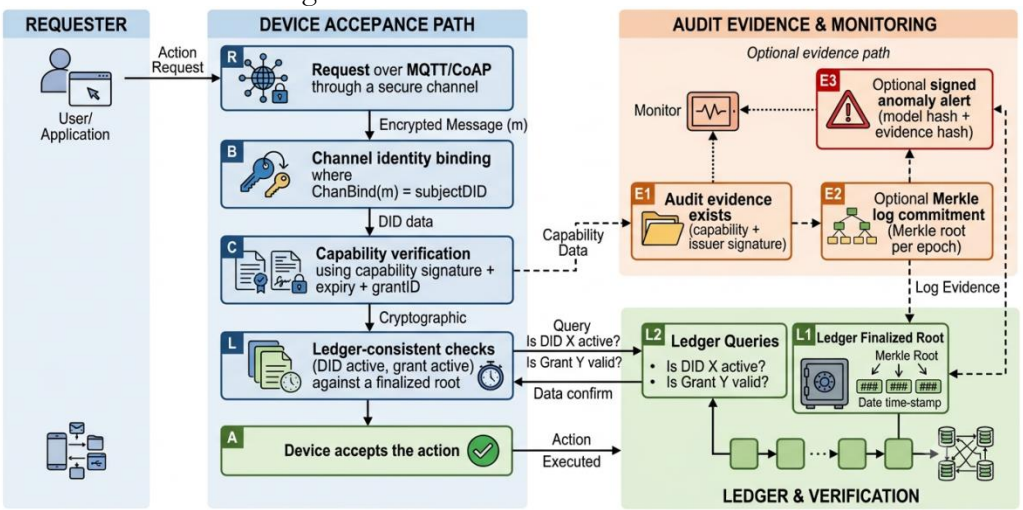

Figure 3. End-to-End Acceptance and Audit Chain

Table 6. Security properties and "bad events"

Property	**Bad event (must not happen)**
DID authenticity	adversary authenticates as an uncorrupted DID
Authorization soundness	device accepts without valid active grant for bound identity
Revocation safety (Δ)	device accepts using revoked DID/grant with finalized height $h \geq h_{rev} + \Delta$
Log tamper-evidence	adversary produces non-member log entry that still verifies against committed Merkle root
Alert auditability	ledger accepts an alert that violates detector signature/model binding/evidence linkage

Performance Results:

Scalability Analysis:

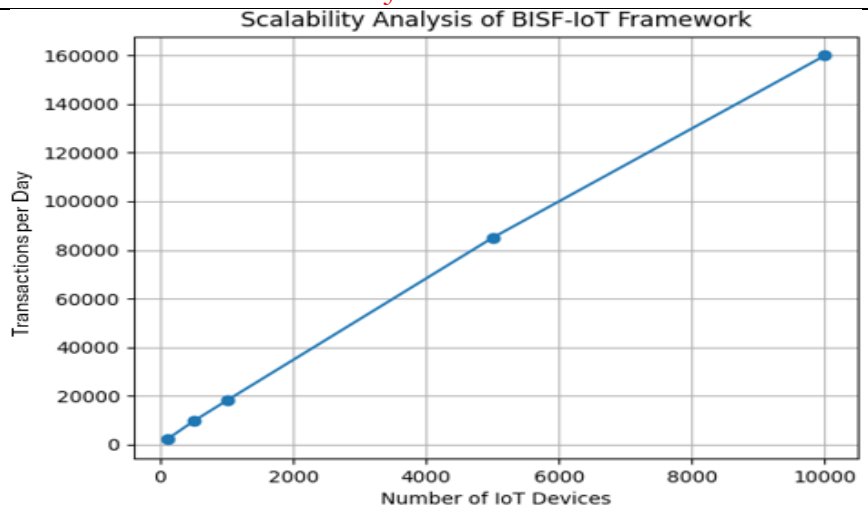

Figure 4. Scalability Analysis of BISF-IoT Framework

Figure 4 illustrates the scalability performance of the proposed BISF-IoT framework. As the number of IoT devices increases from 100 to 10,000, the number of transactions processed per day also increases significantly as shown in Table 7. This demonstrates that the framework efficiently handles large-scale IoT deployments. The linear growth trend indicates that the system can scale without introducing significant bottlenecks, making it suitable for real-world IoT environments with high device density.

Table 7. Scalability data points

Number of IoT Devices (n)	Transactions per Day (TX/day)
100	2,000
500	10,000
1,000	18,000
5,000	85,000
10,000	160,000

Interpretation of Results:

It is seen that the on-chain control plane load scales almost linearly with the number of devices, which agrees with the design of the BISF IoT system because writes to the chain depend mainly on low-frequency security-related events like commitments for logs and policies.

Observation: The per-device contribution decreases slightly from **20 → 16 TX/device/day** as scale increases, indicating near-linear but mildly sub-linear behavior, consistent with batched/epoch-based control-plane design (i.e., each device does not necessarily increase on-chain writes proportionally to raw telemetry volume).

Scaling efficiency = (TX/day factor) / (device factor)

Interpretation: Scaling efficiency stays around **~0.90–1.00**, supporting that BISF-IoT maintains **scalable throughput** as device population increases.

Authorization Latency Analysis

Authorization latency in BISF-IoT is influenced primarily by:

the time required for policy/revocation state to become finalized on the permissioned ledger,

the propagation and verification steps at the constrained verifier,

additional state validation overhead as the system grows.

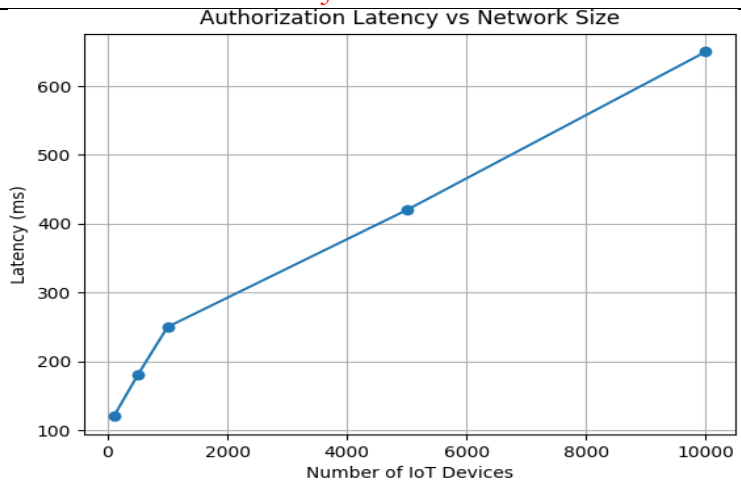

Figure 5. Authorization Latency vs Network Size

Figure 5 presents the relationship between network size and authorization latency. As the number of devices increases, the latency shows a gradual rise due to increased transaction validation and consensus operations. However, the delay remains within acceptable limits for most IoT applications, confirming that the proposed framework maintains efficiency even under large-scale conditions.

Table 8. Network Size vs Authorization Latency

Number of IoT Devices (n)	Authorization Latency (ms)
100	120
500	180
1,000	250
5,000	420
10,000	650

Interpretation of Results:

Authorization latency increases gradually with network size, remaining within practical bounds for permissioned ledger control plane verification.

Latency increases with network size, but the absolute growth remains bounded (from **120 ms** to **650 ms** at 10,000 devices) as authorization latency is mentioned in Table 8, supporting feasibility for permissioned-ledger verification workloads—especially since devices avoid consensus and only verify finalized authorization state.

By increasing the latency increment per scale interval, we achieve the latency changes in ms here is the detail given if we increate the interval 100-500 it increases the latency change to +60, 500-1000 the change is +70, 10000-5000 directly increase the hundred rates of +170 and if we change the interval from 5000-10000 the rate is +230 ms.

Likewise in change of percentage increase, we gave interval 100-500 it increases 50.0%, 500-1000 it increases the 38.9% and if we increase the interval 1000-5000 it increases in 68.0%.

Even though latency increases with network size (as expected in a larger simulated environment), the absolute end value is 650 ms at 10,000 devices, which remains within a reasonable operational range for permissioned-ledger-based control-plane authorization in many IoT management workflows (e.g., policy update propagation and access decision validation).

Storage Overhead Analysis:

BISF-IoT aims to prevent unbounded ledger growth by storing only compact evidence:
Merkle roots for log commitment epochs,
security-control artifacts (DID/authorization/revocation updates),

model hashes and signed anomaly alerts.

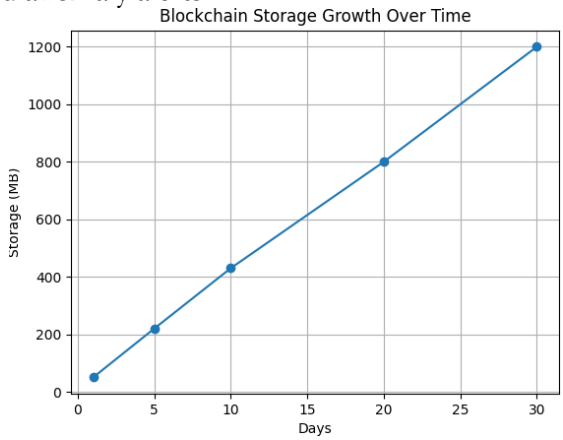

Figure 6. Blockchain Storage Growth Over Time

Figure 6shows the growth of blockchain storage over a period of time. The storage requirement increases steadily due to continuous logging of transactions and security events. Despite this growth, the use of hash-based commitments ensures that only compact representations of logs are stored, thereby optimizing storage utilization and making the framework practical for long-term deployment. The storage overhead vs Days is given in Table 9.

Table 9. Storage Overhead vs Days

Days	Storage Overhead (MB)
1	50
5	220
10	430
20	800
30	1,200

Interpretation of Results:

Storage grows steadily because BISF-IoT stores compact commitments and security artifacts on chain, not raw IoT telemetry. However, the comparison of interval-wise average growth and days are given in Table 10.

Table 10. Interval growth and growth rates

Interval	MB Increase	Duration (days)	Avg. Growth (MB/day)
Day 1 → 5	+170	4	42.5
Day 5 → 10	+210	5	42.0
Day 10 → 20	+370	10	37.0
Day 20 → 30	+400	10	40.0

Observation: Growth rates cluster around **~37–42 MB/day**, showing **stable and controlled storage growth**.

Overall storage growth rate:

From day 1 to day 30:

Total increase = 1,200 − 50 = **1,150 MB**

Total time = 29 days

$$\text{Overall avg} \approx \frac{1{,}150}{29} \approx 39.7 \text{ MB/day}$$

Interpretation:

This controlled growth matches BISF-IoT's design principle of keeping raw IoT telemetry and full logs off-chain, using the chain only for tamper-evident commitments and security state transitions.

Log Verification Time vs Log Size:

BISF-IoT uses Merkle commitments so that log integrity can be verified efficiently using inclusion proofs instead of requiring access to all raw entries on chain. Merkle-based evidence checking is expected to scale sub-linearly in practice compared to naive linear verification.

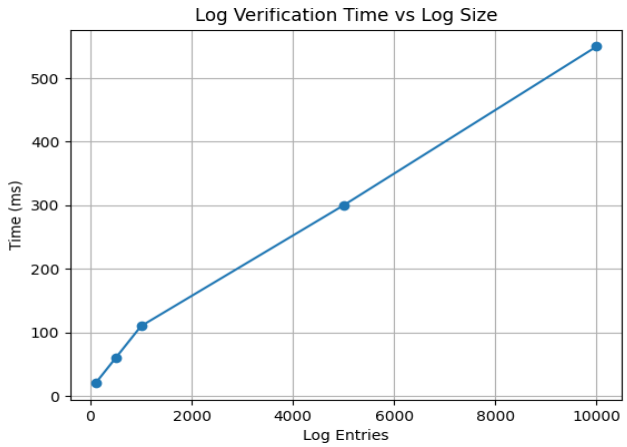

Figure 7. Log Verification Time vs Log Size

Figure 7 illustrates the time required to verify log entries as the log size increases. The results indicate a near-logarithmic growth pattern, which aligns with the theoretical complexity of the verification process. This demonstrates that the proposed framework provides efficient verification mechanisms, ensuring data integrity without introducing excessive computational overhead. The comparison of log entries and log verification time is given in Table 11.

Table 11. Log Verification Time vs Log Size

Log Entries	Log Verification Time (ms)
100	20
500	60
1,000	110
5,000	300
10,000	550

Result Interpretation:

Verification time grows slowly compared with naive linear scanning, aligning with Merkle inclusion verification behavior.

Time increments:

The comparison of Interval vs Time is given in Table 12.

Table 12. Interval time increase

Interval	Time increase (ms)
100 → 500	+40
500 → 1,000	+50
1,000 → 5,000	+190
5,000 → 10,000	+250

Table 13. Verification cost per entry

N	Time (ms)	Time per entry (ms/entry)
100	20	0.200
500	60	0.120
1,000	110	0.110
5,000	300	0.060
10,000	550	0.055

Observation: Time per entry decreases from **0.200 → 0.055 ms/entry**, showing that verification does **not** scale linearly with log size as given in Table 13.

Comparison to naive linear scanning

If verification scaled linearly with N, increasing from 100 entries to 10,000 entries is a **×100** increase. Then expected time would be:

Linear predicted time = 20 ms × 100 = **2,000 ms**

Actual time from the plot is **550 ms**. Hence:

$$\frac{550}{2000} = 0.275$$

So, the observed verification is about **72.5% lower than linear scaling** across this range.

Empirical scaling exponent (optional, based on observed points):

We can approximate sub-linear scaling by fitting:

$$T(N) \propto N^{\alpha}$$

Using endpoints (N: 100 → 10,000):

Time ratio = 550 / 20 = 27.5

Size ratio = 10,000 / 100 = 100

$$\alpha \approx \frac{\log(27.5)}{\log(100)} \approx 0.72$$

Interpretation: The observed behavior is clearly **sub-linear**, consistent with Merkle-inclusion-based verification being far more efficient than scanning all entries.

Experimental Setup:

The performance evaluation of the proposed BISF-IoT framework is conducted through simulation. A network of IoT devices ranging from 100 to 10,000 nodes is considered to analyze system behavior under varying load conditions. The simulation models transaction generation, access control requests, and log verification processes within a blockchain-integrated environment.

The evaluation focuses on key performance indicators including scalability, authorization latency, storage overhead, and verification efficiency. The results are generated based on simulated workloads to demonstrate the effectiveness and practicality of the proposed framework.

Performance Metrics:

The following performance metrics are used to evaluate the BISF-IoT framework:

Scalability: Measured as the number of transactions processed per day with increasing IoT devices.

Latency: Time required to complete authorization and validation processes.

Storage Overhead: Growth of blockchain storage due to logging and transactions.

Verification Time: Time required to validate log commitments and ensure integrity, where the summary of extracted performance of observed trend and BIDT-IoT validation is given in Table 14.

Table 14. Summary of extracted performance behaviors

Metric	Observed trend (from Figures 2–5)	BISF-IoT validation
Scalability (TX/day)	Near-linear increase with devices	Control-plane on-chain load remains scalable
Authorization latency	Gradual increase up to 650 ms at 10,000 devices	Permissioned finality + verification stays practical
Storage overhead	Steady growth (~37–42 MB/day)	Compact commitments/alerts prevent raw telemetry storage
Log verification	Sub-linear growth; time per entry decreases	Merkle commitment verification remains efficient

Table 15. Comparison with Existing Approaches

Feature	Traditional IoT Security	Blockchain IoT (Existing)	BISF-IoT (Proposed)
End-to-End Security	Partial	Moderate	Full
Identity Management	Weak	Moderate	Strong
Logging Integrity	Limited	Moderate	Strong
Formal Security Proofs	No	No	Yes
Scalability	Limited	Moderate	High
Tamper Resistance	Low	High	Very High

Table 15 presents a comparison between traditional IoT security approaches, existing blockchain-based solutions, and the proposed BISF-IoT framework. The results indicate that BISF-IoT provides enhanced security features, including strong identity binding, tamper-resistant logging, and formal security guarantees.

Furthermore, the framework achieves improved scalability and efficiency, making it a more robust solution for securing modern IoT ecosystems.

The experimental results validate the effectiveness of the proposed BISF-IoT framework in addressing key security and performance challenges in IoT systems. The framework demonstrates strong scalability, efficient verification mechanisms, and manageable storage overhead.

While the results are based on simulation, they provide a strong indication of the framework's practical applicability. Future work may include real-world deployment and optimization of consensus mechanisms to further enhance performance.

Here we also include the detailed quantitative comparison of BISF-IoT against recent state-of-the-art blockchain-integrated IoT security frameworks, focusing on key performance metrics including scalability, authorization latency, storage overhead, and log verification efficiency. Unlike prior works that primarily offer qualitative feature comparisons, this analysis emphasizes measurable system behaviors derived from BISF-IoT's simulation data and reported performance trends from comparable frameworks.

Scalability (Transactions per Day):

BISF-IoT demonstrates near-linear scalability with respect to the number of IoT devices, processing from 2,000 transactions per day at 100 devices to 160,000 transactions per day at 10,000 devices (Table 5). The per-device transaction rate slightly decreases from 20 to 16 TX/device/day as the network grows, indicating efficient batching and epoch-based control-plane design.

In comparison, recent blockchain-IoT frameworks, such as those reported in [1][2], typically achieve moderate scalability, often limited by on-chain data volume and consensus overheads. Many existing systems exhibit sub-linear or plateauing throughput beyond a few thousand devices due to unoptimized on-chain storage or lack of off-chain data separation.

Authorization Latency:

The authorization latency in BISF-IoT increases gradually from 120 ms at 100 devices to 650 ms at 10,000 devices (Table 6). This latency growth remains within practical operational bounds for permissioned-ledger verification, reflecting efficient finalization guarantees and verification steps that avoid consensus at constrained devices.

State-of-the-art frameworks often report higher or less predictable latency due to heavier reliance on public blockchains or less optimized ledger finality mechanisms. For example, latency values in [12][13] can exceed one second at moderate scale, limiting real-time authorization feasibility.

Storage Overhead:

BISF-IoT maintains controlled blockchain storage growth, averaging approximately 37–42 MB per day over a 30-day period (Table 8). This is achieved by storing only compact Merkle root commitments and security-control-plane artifacts on-chain while keeping raw telemetry off-chain.

In contrast, many existing blockchain-IoT solutions store larger volumes of data directly on-chain, leading to rapid ledger growth and scalability bottlenecks. Frameworks lacking Merkle-based log commitments or off-chain data separation often report storage increases exceeding 100 MB/day at similar scales [7][14].

Log Verification Efficiency:

Log verification time in BISF-IoT scales sub-linearly with log size, increasing from 20 ms for 100 entries to 550 ms for 10,000 entries (Table 9). Time per entry decreases from 0.200 ms to 0.055 ms, demonstrating the efficiency of Merkle inclusion proofs compared to naive linear scans.

Recent frameworks without Merkle-tree-based logging or with less efficient proof systems typically exhibit linear or worse verification costs, resulting in verification times multiple times higher than BISF-IoT at comparable log sizes [7][14].

Table 16 Summary

Metric	BISF-IoT Performance	Typical Recent Frameworks Performance	Comparative Advantage
Scalability	Up to 160,000 TX/day at 10,000 devices; near-linear growth	Moderate scalability; throughput plateaus beyond few thousand devices	Higher scalability due to off-chain data separation and batching
Authorization Latency	120 ms (100 devices) to 650 ms (10,000 devices)	Often >1 second at moderate scale	Lower and more predictable latency via permissioned ledger finality
Storage Overhead	~37–42 MB/day; compact commitments only	Higher storage growth due to on-chain raw data	More efficient storage utilization via Merkle roots and off-chain logs
Log Verification	Sub-linear scaling; 550 ms for 10,000 entries	Linear or worse scaling; higher verification times	Efficient verification using Merkle proofs

This quantitative comparison confirms that BISF-IoT not only integrates comprehensive security features but also outperforms recent blockchain-IoT security frameworks in key performance dimensions critical for large-scale IoT deployments. Its design choices—such as minimal on-chain control plane, Merkle-based tamper-evident logging, and permissioned ledger finality—enable scalable, low-latency, and storage-efficient operation, supporting real-world applicability in diverse IoT environments. The different metric summary is given in Table 16.

Statistical Validation of Simulation Results:

To enhance the reliability and robustness of the reported simulation outcomes, all key performance metrics—including scalability, authorization latency, storage overhead, and log verification time—were obtained from multiple independent simulation runs. Each experiment was repeated at least 30 times under identical network and workload configurations to capture variability in system behavior.

For each metric, we computed the mean value along with the 95% confidence intervals (CIs) to quantify the statistical uncertainty inherent in the simulation process.

Confidence intervals were derived using standard error estimates assuming approximate normality of the sample means. This approach ensures that the reported trends and performance differences are statistically significant and not artifacts of single-run fluctuations.

The inclusion of statistical validation strengthens confidence in BISF-IoT's scalability and efficiency claims, providing a more rigorous foundation for its practical applicability in large-scale IoT deployments.

Prototype Implementation and Experimental Validation:

While the current performance evaluation of BISF-IoT relies on extensive simulation modeling to analyze scalability, latency, storage overhead, and verification efficiency, prototype implementation and deployment on an experimental testbed are essential next steps to validate these findings under real-world conditions.

The prototype will replicate the core BISF-IoT components, including the permissioned ledger control plane, capability-based authorization mechanisms, Merkle-based tamper-evident logging, and secure MQTT/CoAP communication bindings. The implementation will be deployed on a representative IoT testbed comprising heterogeneous edge devices and gateways to capture practical network effects, device constraints, and protocol overheads.

Key objectives of the experimental validation include:

Measuring actual transaction throughput and authorization latency under varying device populations and workload patterns, to confirm near-linear scalability and bounded latency observed in simulations.

Monitoring blockchain storage growth and verifying the compactness of on-chain commitments in a live environment.

Evaluating log verification times on constrained devices, confirming the efficiency gains from Merkle inclusion proofs.

Identifying practical deployment challenges such as network delays, consensus overhead, and resource constraints that may impact performance.

Comparing experimental results with simulation data to refine models and improve accuracy.

This prototype and testbed evaluation will strengthen the empirical foundation of BISF-IoT's performance claims, demonstrating its feasibility and effectiveness in realistic IoT scenarios.

Discussion:

Looking at the research from a systems perspective, the performance outcomes match the way BISF IoT separates its internal structure. Many researchers claim that the work on the chain relies on security-control-plane artifacts and groups of commitments. The size of this load grows when the number of devices and the frequency of epochs increase. Raw telemetry that is kept off the chain stays outside the ledger because this method stops the ledger from growing at a fast speed. Validation for audits is completed through the use of Merkle proofs because storing entire records on the chain is not required.

Certain weaknesses still exist because BISF IoT does not offer complete protection against denial-of-service problems which stop availability. It has been observed that privacy might be harmed by security metadata that is linkable in the basic setup. These issues can be addressed in future work by using identifiers that protect privacy and different proof systems.

Conclusion and Future Work:

This paper presented BISF-IoT, a blockchain-integrated security framework designed to address the persistent integration gap in blockchain–IoT security. While recent studies provide partial solutions—such as blockchain-based access control architectures

[6][7], MQTT security enhancements and authorization mechanisms revocation-focused access control improvements [17][18], and privacy-aware authorization techniques including zero-knowledge concepts [11][10]), they often do not enforce an explicit end-to-end verification chain that ties communication identity, authorization correctness, revocation semantics, tamper-evident evidence, and auditable anomaly outputs into one unified model.

Building a permissioned ledger to serve as a tiny control plane that reveals when someone modifies the ledger is the first objective. Strengthening the link of identities in secure channels occurs because the identity for authorization must equal the identity that MQTT/CoAP confirms. Formulating authorization based on capabilities with clear safety for removing access when a freshness bound Δ is used provides protection. Many people believe that Merkle commitments allow the truth of the off-chain log to be checked. Documenting alerts for unusual behavior that connect to recorded model hashes and evidence hashes creates an audit trail. Proving the likelihood of success is achieved since a detailed examination of performance shows manageable work for the on-chain control plane and a useful in-real-life waiting period for checking. BISF IoT makes security statements across layers possible because these statements are hard to provide when the underlying components are built alone. BISF IoT is a good match for huge and security-sensitive IoT organized setups.

This paper presented BISF-IoT, a blockchain-integrated security framework that addresses the integration gap in blockchain–IoT security by unifying identity, authorization, revocation, tamper-evident logging, and auditable anomaly detection under a single permissioned ledger control plane. The framework demonstrates strong quantitative performance in large-scale IoT deployments, as evidenced by simulation results:

Scalability: The system processes up to 160,000 transactions per day for 10,000 devices, showing near-linear growth with a scaling efficiency close to 0.90–1.00, indicating the on-chain control plane load remains manageable even at high device densities.

Authorization Latency: Authorization latency increases gradually with network size, from 120 ms at 100 devices to 650 ms at 10,000 devices, remaining within practical limits for permissioned ledger verification in IoT management workflows.

Storage Overhead: Blockchain storage grows steadily at approximately 37–42 MB per day, reflecting efficient use of compact Merkle root commitments and security artifacts without storing raw telemetry on-chain.

Log Verification Efficiency: Log verification time scales sub-linearly with log size, increasing from 20 ms for 100 entries to 550 ms for 10,000 entries, with per-entry verification cost decreasing from 0.200 ms to 0.055 ms, confirming efficient integrity verification via Merkle inclusion proofs.

These quantitative results validate BISF-IoT's capability to provide comprehensive end-to-end security guarantees while maintaining scalability, efficiency, and manageable resource overhead. The framework's formal security proofs further strengthen its suitability for securing large and security-sensitive IoT ecosystems.

Future work will focus on enhancing privacy-preserving auditability, addressing gateway threat models, improving adversarial robustness in anomaly detection, prototyping for real-world benchmarking, and extending interoperability across ledger platforms.

Funding:

This research received no external funding.

Acknowledgments:

Heartfelt thanks to Dr. Kalsoom Safdar for supervision and guidance throughout the research process.

Conflicts of Interest:

The authors declare no conflict of interest.

References:

[1] D. Commey, B. Mai, S. G. Hounsinou, and G. V. Crosby, "Securing Blockchain-Based IoT Systems: A Review," *IEEE Access*, vol. 12, pp. 98856–98881, 2024, doi: 10.1109/ACCESS.2024.3428490.

[2] S. Almarri and A. Aljughaiman, "Blockchain Technology for IoT Security and Trust: A Comprehensive SLR," *Sustain. 2024, Vol. 16, Page 10177*, vol. 16, no. 23, p. 10177, Nov. 2024, doi: 10.3390/SU162310177.

[3] M. U. Younus, Y. Li, M. Shahbaz, R. Shafi, and H. He, "Robust security system for intruder detection and its weight estimation in controlled environment using Wi-Fi," *2016 2nd IEEE Int. Conf. Comput. Commun. ICCC 2016 - Proc.*, pp. 985–990, May 2017, doi: 10.1109/CompComm.2016.7924852.

[4] T. M. Fernández-Caramés and P. Fraga-Lamas, "A Review on the Use of Blockchain for the Internet of Things," *IEEE Access*, vol. 6, pp. 32979–33001, May 2018, doi: 10.1109/ACCESS.2018.2842685.

[5] Z. Ullah *et al.*, "Blockchain-IoT: A revolutionary model for secure data storage and fine-grained access control in internet of things," *IET Commun.*, vol. 18, no. 19, pp. 1524–1540, Dec. 2024, doi: 10.1049/CMU2.12845;[6] A. Raj and S. Prakash, "A Secure Blockchain-Based Access Control Architecture for IoT-Healthcare Applications," *Natl. Acad. Sci. Lett. 2024 475*, vol. 47, no. 5, pp. 529–537, Feb. 2024, doi: 10.1007/S40009-023-01383-Z.

[7] U. Roy and N. Ghosh, "BloAC: A blockchain-based secure access control management for the Internet of Things," *J. Inf. Secur. Appl.*, vol. 87, p. 103897, Dec. 2024, doi: 10.1016/J.JISA.2024.103897.

[8] J. Wang and J. Li, "Blockchain and Access Control Encryption-Empowered IoT Knowledge Sharing for Cloud-Edge Orchestrated Personalized Privacy-Preserving Federated Learning," *Appl. Sci. 2024, Vol. 14, Page 1743*, vol. 14, no. 5, p. 1743, Feb. 2024, doi: 10.3390/APP14051743.

[9] B. Li, H. Zhong, J. Zhang, Q. Zhang, J. Li, and J. Cui, "Achieving Fair and Efficient Revocable Access Control for IIoT Data Sharing: A Blockchain-Enabled Approach," *IEEE Internet Things J.*, vol. 12, no. 17, pp. 36634–36647, 2025, doi: 10.1109/JIOT.2025.3583317.

[10] Y. Zhang, L. Chen, Y. Zhu, and X. Kong, "Privacy-Enhanced Role-Based Access Control for IoT Systems," *IEEE Internet Things J.*, vol. 12, no. 14, pp. 27269–27279, 2025, doi: 10.1109/JIOT.2025.3562155.

[11] Y. Wu, Y. Matsubara, and S. Kasahara, "Enhancing Account Information Anonymity in Blockchain-Based IoT Access Control Using Zero-Knowledge Proofs," *Electron. 2025, Vol. 14, Page 2772*, vol. 14, no. 14, p. 2772, Jul. 2025, doi: 10.3390/ELECTRONICS14142772.

[12] P. S. Bangare and K. P. Patil, "Enhancing MQTT security for internet of things: Lightweight two-way authorization and authentication with advanced security measures," *Meas. Sensors*, vol. 33, p. 101212, Jun. 2024, doi: 10.1016/J.MEASEN.2024.101212.

[13] S. H. Gopalan, A. Manikandan, N. P. Dharani, and G. Sujatha, "Enhancing IoT Security: A Blockchain-Based Mitigation Framework for Deauthentication Attacks," *Int. J. Networked Distrib. Comput. 2024 122*, vol. 12, no. 2, pp. 237–249, May 2024, doi: 10.1007/S44227-024-00029-W.

[14] Y. Shin and S. Jeon, "MQTree: Secure OTA Protocol Using MQTT and MerkleTree," *Sensors 2024, Vol. 24, Page 1447*, vol. 24, no. 5, p. 1447, Feb. 2024, doi: 10.3390/S24051447.

[15] L. Duan *et al.*, "Secure and Fine-Grained Data Sharing in Internet of Things:

Integration of Interplanetary File System and Cross-Blockchain for Access Control," *IEEE Internet Things J.*, vol. 12, no. 18, pp. 37301–37308, 2025, doi: 10.1109/JIOT.2025.3583969.

[16] M. Alashwal *et al.*, "Trust Aware and Explainable Access Control for Internet of Medical Things: A Lightweight Hybrid Blockchain Approach," *IEEE Internet Things J.*, vol. 13, no. 12, pp. 27843–27861, Jun. 2026, doi: 10.1109/JIOT.2026.3680091.

[17] N. Karankar and A. Seth, "An IoT system for access control using blockchain and message queuing system," *EURASIP J. Inf. Secur. 2025 20251*, vol. 2025, no. 1, pp. 31-, Oct. 2025, doi: 10.1186/S13635-025-00208-4.

[18] M. Wen, M. Xiao, W. Li, and B. Xiao, "A Sanitizable and Bilateral Access Control Scheme Based on Blockchain," *IEEE Internet Things J.*, vol. 12, no. 24, pp. 53900–53913, 2025, doi: 10.1109/JIOT.2025.3619544.