ZK-FLGuard: Verifiable Privacy via Zero-Knowledge Proofs in Federated Anomaly Detection for 5G Edge-IoT Systems

Manuel J. C. S. Reis

Department of Engineering and IEETA, University of Trás-os-Montes e Alto Douro, Vila Real, Portugal
Email: mcabral@utad.pt (M.J.C.S.R.)
Manuscript received June 15, 2025; revised August 8, 2025; accepted November 25, 2025; published January 25, 2026

***Abstract*—This paper presents Zero-Knowledge Federated Learning Guard (ZK-FLGuard), a privacy-preserving and verifiable federated learning framework for real-time anomaly detection in Fifth-Generation Mobile Network (5G)-enabled Internet of Things (IoT) environments. Building on the integration of zero-knowledge proofs (zk-SNARK—Zero-Knowledge Succinct Non-interactive Argument of Knowledge) and blockchain-based access control, ZK-FLGuard ensures the integrity of model updates without exposing private data. Using real-world intrusion detection datasets (CICIDS2017—Canadian Institute for Cybersecurity Intrusion Detection System 2017, TON_IoT—Telecommunications Organisation of the National Security—IoT) and a synthetic adversarial dataset, our evaluation shows that ZK-FLGuard achieves up to 0.96 F1-score (harmonic mean of precision and recall), improves recall in low-frequency attack detection, and introduces less than 10% additional latency overhead compared to standard Federated Learning (FL). Compared with centralized Long Short-Term Memory (LSTM) and FL without Zero-Knowledge Proof (ZKP), ZK-FLGuard provides competitive accuracy while ensuring verifiable computation and strong privacy guarantees. We address the critical challenge of securing federated anomaly detection in 5G-enabled IoT systems against data leakage, model poisoning, and unauthorized access. While FL preserves privacy by keeping raw data local, it remains vulnerable to gradient leakage and adversarial manipulation. Our hypothesis is that combining zero-knowledge proofs and blockchain with FL can deliver a scalable, tamper-resistant, and privacy-preserving detection pipeline suitable for resource-constrained edge environments.**

***Keywords*—federated learning, Zero-Knowledge Proofs (ZKP), edge computing, 5G networks, Internet of Things (IoT), anomaly detection, privacy-preserving machine learning**

I. Introduction

The proliferation of 5G networks and Internet of Things (IoT) devices has ushered in an era of ultra-connected, latency-sensitive applications, including smart cities, e-health, and industrial IoT. However, this interconnectedness also amplifies the cybersecurity threat surface, with attackers exploiting the massive data generation and decentralized architecture inherent in these systems [1, 2].

To ensure real-time anomaly detection and threat resilience in such dynamic environments, Federated Learning (FL) has emerged as a promising paradigm. FL allows collaborative training across edge devices without sharing raw data, thus preserving user privacy and enabling local intelligence [3, 4]. However, despite these advantages, FL remains vulnerable to security and privacy threats—including gradient leakage, model poisoning, and backdoor attacks—which can occur even without direct access to raw training data [5, 6].

To mitigate these concerns, researchers have proposed incorporating cryptographic mechanisms into FL frameworks. Among them, Zero-Knowledge Proofs (ZKPs) stand out for their ability to verify the correctness of computations without revealing the underlying data [7–9]. In FL settings, ZKPs enable each client to prove that its model update was derived from legitimate computations on authorized data, thus providing a formal basis for tamper-proof and privacy-preserving learning [10].

Moreover, combining FL with blockchain introduces transparent, tamper-resistant record-keeping of contributions, access control, and trust management across semi-trusted devices. While hybrid blockchain architectures have been explored to balance throughput and decentralization in IoT networks [11, 12], there is still no unified architecture that integrates ZKPs, blockchain, and federated learning into a scalable, edge-compatible solution for real-time anomaly detection.

This work addresses that gap by introducing Zero-Knowledge Federated Learning Guard (ZK-FLGuard), a privacy-preserving and verifiable federated anomaly detection framework designed for 5G-enabled IoT ecosystems. Unlike existing approaches, ZK-FLGuard combines FL, zk-SNARK-based proof-of-update, and blockchain identity management into a cohesive system that both safeguards privacy and ensures computational integrity. Our hypothesis is that this combination can achieve competitive detection accuracy while maintaining verifiable trust and low system overhead in resource-constrained environments.

The main contributions of this work are:

1. A hybrid architecture that integrates FL, zk-SNARK-based proof-of-update, and blockchain identity management;
2. A formal privacy analysis demonstrating negligible leakage under zk-SNARK assumptions;
3. Implementation and evaluation on CICIDS2017, TON_IoT, and an adversarial synthetic dataset;
4. Comparative performance analysis showing minimal overhead with competitive accuracy.

Across these evaluations, ZK-FLGuard achieved F1-scores of up to 0.96, maintained accuracy above 0.94, and reduced bandwidth usage by over 30% compared to full model synchronization—all while keeping additional latency under 10%. These results highlight the framework's robustness, scalability, and suitability for securing federated intelligence in next-generation smart environments.

II. Related Work

A. Federated Learning in IoT and Anomaly Detection

Recent years have witnessed a surge in applying Federated Learning (FL) to decentralized systems, particularly IoT-based anomaly detection. FL offers privacy benefits by

keeping raw data on-device, but several works have shown susceptibility to gradient leakage [5], model poisoning [6], and difficulty in handling non-IID data [4]:

- Kairouz *et al.* [3] presented a foundational survey on FL challenges and architectures.
- Hard *et al.* [13] explored FL for mobile keyboard predictions, highlighting communication bottlenecks.
- Geyer *et al.* [14] introduced semi-supervised FL techniques in constrained IoT settings.
- Li *et al.* [15] applied FL to real-time edge anomaly detection using LSTM architectures.
- Reis [16] proposed Edge-FLGuard, an FL-based framework for anomaly detection in IoT with communication-efficient edge training.

Recent approaches such as RFBDS/PrivRFBDS [17] and SAFELearning [18] offer robust aggregation mechanisms and targeted defense against backdoor injection in FL, highlighting the growing demand for integrity verification under adversarial conditions.

However, these works either lack formal cryptographic guarantees, rely on heuristic anomaly filtering, or do not consider the computational constraints of heterogeneous IoT edge nodes. None of them integrate a verifiable proof mechanism to ensure that only legitimate updates influence the global model. This absence of verifiable computation motivates our integration of Zero-Knowledge Proofs (ZKPs) into the FL pipeline for secure anomaly detection in 5G-enabled IoT environments.

While these contributions advance decentralized learning, most either lack formal cryptographic guarantees, focus on a single security mechanism, or fail to address the computational constraints of heterogeneous IoT edge devices. To highlight these limitations, Table 1 provides a comparative overview of representative recent works, evaluating them along six key dimensions: use of FL, integration of ZKPs, blockchain support, IoT/5G applicability, anomaly detection capability, and edge compatibility. As the table shows, ZK-FLGuard is the only framework that combines all six features into a unified, verifiable, and edge-deployable architecture.

Table 1. Summary of recent works integrating federated learning, privacy mechanisms, and blockchain technologies for IoT/5G security. ZK-FLGuard is the only framework combining all key components for secure, real-time, edge-based anomaly detection

Work	FL	ZKP	Blockchain	IoT/5G	Anomaly Detection	Edge Compatibility
[3]	✓	✗	✗	Partial	✗	✗
[19]	✓	✓	✓	✗	Partial	✗
[20]	✓	✗	✓	✓	✓	✓
[16]	✓	✗	✗	✓	✓	✓
ZK-FLGuard (This Work)	✓	✓	✓	✓	✓	✓

B. Zero-Knowledge Proofs in Secure Computation

ZKPs provide formal privacy guarantees in distributed systems, allowing one party to prove the correctness of a computation without revealing inputs. Widely adopted in blockchain and privacy-preserving protocols, ZKPs are less explored in federated learning pipelines:

- Ben-Sasson *et al.* [7] introduced zk-SNARK, enabling scalable ZKPs in real-world applications.
- Groth [9] developed short non-interactive ZKPs that support efficient cryptographic proofs.
- Bünz *et al.* [21] proposed Bulletproofs for efficient range proofs in cryptocurrency systems.

Recent work has suggested combining FL with ZKP for secure updates, but such efforts are preliminary and lack comprehensive evaluation in 5G/IoT edge settings [19]. Petrosino *et al.* [22] applied zk-SNARK to federated learning in a healthcare setting, confirming their feasibility on edge devices and validating security guarantees in regulated environments. Similarly, Xing *et al.* [23] proposed PZKP-FL, a practical framework integrating ZKP with blockchain to secure FL pipelines, closely aligning with the goals of this work.

Despite these advances, existing ZKP-FL integrations either target narrow application domains (e.g., healthcare) or neglect performance validation under adversarial and resource-constrained edge scenarios. Furthermore, they often omit a decentralized trust layer, which limits scalability in heterogeneous IoT ecosystems. These shortcomings justify the need for ZK-FLGuard's unified and performance-tested design.

C. Blockchain for Authentication in IoT

Blockchain has been proposed as a decentralized alternative for identity verification, access control, and data integrity in IoT networks:

- Abdelmaboud *et al.* [24] reviewed blockchain-based security in IoT and proposed hybrid frameworks.
- Dorri *et al.* [25] presented lightweight blockchain solutions for smart homes.
- Xue *et al.* [26] explored scalability and convergence in blockchain-IoT integrations.
- Reis [20] introduced a blockchain-enhanced security framework for 5G edge computing in IoT, which inspires the trust model used in ZK-FLGuard.

A recent survey by Jiang *et al.* [27] offers a comprehensive taxonomy of blockchain-integrated FL models in IoT, including access control, secure aggregation, and incentive mechanisms—reinforcing the relevance of distributed ledger technologies in this space.

While blockchain enhances trust, most IoT-oriented implementations suffer from latency and throughput limitations when scaled, and they lack integration with cryptographically verifiable FL mechanisms. This motivates our use of a permissioned blockchain layer to handle identity and logging without introducing significant overhead.

D. Comparative Analysis of Existing Frameworks

To better contextualize the contributions of ZK-FLGuard, Table 1 presents a comparative analysis of key related works across multiple dimensions, including Federated Learning (FL), Zero-Knowledge Proofs (ZKP), blockchain integration, and applicability to IoT/5G anomaly detection at the edge.

As we have seen, several studies attempt to integrate FL and blockchain (e.g., EdgeChain, FLChain), and others include lightweight anomaly detection models. However, the lack of:

- Formal proofs of update integrity,
- ZKP-based integrity verification, and
- Privacy-preserving anomaly detection at the edge

remains a gap. ZK-FLGuard explicitly addresses all three

shortcomings by integrating zk-SNARK-based verification, permissioned blockchain authentication, and lightweight, edge-compatible anomaly detection into a single, experimentally validated architecture for 5G IoT environments.

III. System Architecture and Design

ZK-FLGuard is designed as a modular framework integrating federated learning, ZKPs, and blockchain-based identity management to enable real-time, privacy-preserving anomaly detection across 5G-enabled IoT systems.

This section provides a detailed, step-by-step description of the framework, explaining not only the system components but also the design rationale and parameter choices that make ZK-FLGuard suitable for resource-constrained edge environments.

A. Components Overview

The system integrates the following main components:

- IoT Edge Nodes: Edge devices (e.g., Raspberry Pi, Jetson Nano) locally train lightweight anomaly detection models using real-time sensor streams (e.g., temperature, gas, traffic). Each device retains its raw data locally to preserve privacy and participates in collaborative training via gradient updates. We use lightweight LSTM-based models, with parameters such as batch size and learning rate tuned per dataset (see Section V). For example, CICIDS2017 uses a batch size of 64 and learning rate of 0.01, while TON_IoT benefits from smaller batches due to its sensor-stream nature.
- Federated Learning Server: A semi-decentralized coordination layer aggregates model updates using a weighted averaging scheme. This layer can be centralized (cloud-hosted) or distributed using selected edge gateways to support scalability and reduce latency. Weighted aggregation ensures that updates from nodes with more representative data have proportionally greater influence, improving convergence in Non-Independent and Non-Identically Distributed (non-IID) environments.
- Blockchain Identity and Access Control Layer: A permissioned blockchain is used to authenticate participating devices, record update metadata, and manage smart contracts for model versioning. The blockchain ledger ensures tamper-resistance and traceability of all model contributions. Permissioned access minimizes consensus latency while preserving the security guarantees of blockchain-based logging.
- Zero-Knowledge Proof Module: Before sending updates, each client generates a ZKP attesting that its gradient was derived from legitimate, locally authorized data. zk-SNARK (e.g., via ZoKrates) are used to construct non-interactive proofs that are verified by the server before aggregation. The zk-SNARK circuit enforces constraints on data origin and model training steps, ensuring that even a compromised node cannot submit fabricated updates without detection.

B. Communication Flow

The core components of the ZK-FLGuard architecture are summarized below, each playing a distinct role in ensuring secure, privacy-preserving, and verifiable learning across distributed IoT environments:

1. Each authenticated IoT node trains its local model on private data.
2. It computes its model update and generates a zk-SNARK proof of data legitimacy.
3. Both the update and proof are sent to the federated server.
4. The server verifies the proof using a pre-agreed public verification key.
5. Validated updates are aggregated and broadcasted back to the edge nodes.
6. Blockchain logs store metadata for accountability (e.g., node ID, timestamp, hash of update/proof).

This privacy-aware communication scheme ensures that no sensitive data or gradient information is exposed to unauthorized entities. The communication protocol is optimized to reduce overhead: proofs average 2.3 KB, adding negligible bandwidth cost, and verification latency remains under 150 ms per update, ensuring real-time applicability.

The overall structure and interaction between components in ZK-FLGuard are illustrated in Fig. 1. As shown in Fig. 1, the architecture is logically divided into three layers: (1) edge computation and proof generation, (2) federated verification and aggregation, and (3) blockchain-based trust management. This layered separation clarifies the flow of model updates and ensures that computational integrity checks are enforced before any aggregation occurs.

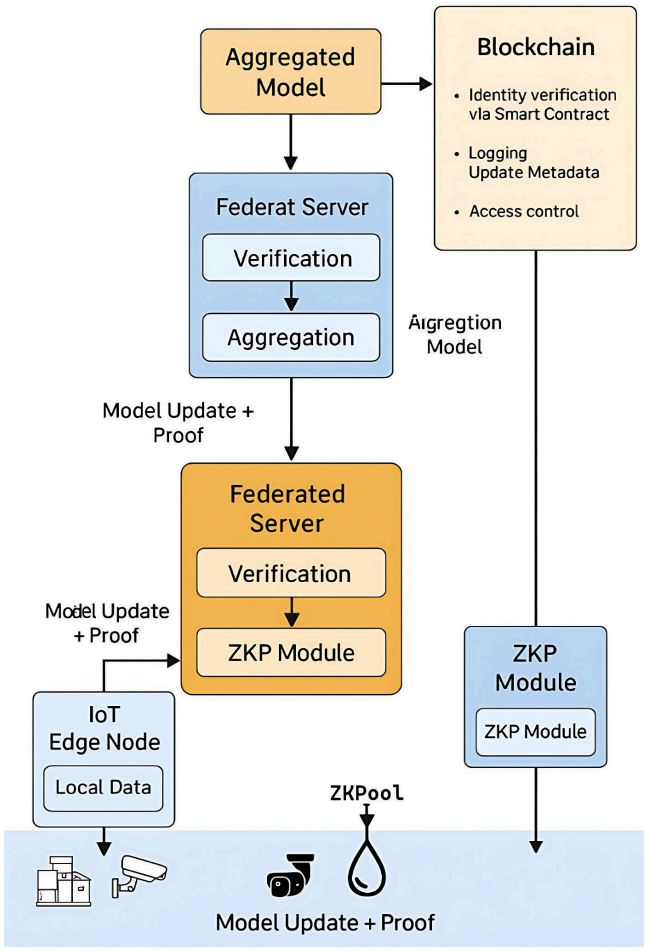

Fig. 1. System-level architecture of ZK-FLGuard. IoT edge nodes locally train anomaly detection models on private data and generate zk-SNARK proofs attesting to update legitimacy. These updates and proofs are sent to the federated server, which first performs proof verification before aggregating valid updates into a global model. The blockchain layer manages identity verification, logs update metadata, and enforces access control via smart contracts. This design ensures that only verifiable, authorized updates

contribute to the aggregated model, while preserving data privacy and supporting accountability.

C. Threat Model

ZK-FLGuard considers a comprehensive adversarial setting that includes:

- White-box attackers: Assume full visibility into model architectures and the ability to reverse-engineer gradients.
- Black-box attackers: Attempt inference or poisoning by manipulating inputs or outputs without model knowledge.
- Malicious clients: Try to inject poisoned or fabricated updates to influence the global model.
- Honest-but-curious servers: Attempt to extract sensitive information from gradient updates or update history.

By leveraging ZKPs, blockchain authentication, and on-device training, the architecture defends against poisoning, backdoor attacks, and gradient inversion with formal, provable guarantees. In particular, the zk-SNARK proof-of-update mechanism prevents any model modification from being accepted unless it can be proven to originate from authorized data and valid computation steps, thus mitigating both stealthy and brute-force adversarial strategies.

IV. Formal Model and Problem Formulation

To formalize the behavior of ZK-FLGuard, we define a three-part model encompassing the federated learning objective, a zero-knowledge-based integrity constraint, and a privacy leakage bound grounded in information theory.

We define the following notation for formal consistency:

- $\mathcal{D}_i$: Local dataset held by edge node i
- $f(\cdot; w_i)$: Local anomaly detection model with weights w_i
- $\mathcal{L}_i(w_i)$: Local loss function evaluated on $\mathcal{D}_i$
- $g_i = \nabla \mathcal{L}_i(w_i)$: Computed gradient update
- π_i: Zero-knowledge proof submitted with update g_i
- w: Global model weights maintained at the coordinator
- α_i: Data proportion weight for client i
- $I(A; B)$: Mutual information between random variables A and B

A. Federated Learning Objective

Consider a set of N IoT clients, where each client $i \in \{1, \dots, N\}$ holds a local dataset $\mathcal{D}_i$. The goal of Federated Learning (FL) is to collaboratively minimize a global objective function across all clients, without sharing raw data.

Let $\ell_i(w)$ denote the local empirical loss of client i on model parameters $w \in R^d$, and let $p_i = \frac{|\mathcal{D}_i|}{\sum_{j=1}^{N} |\mathcal{D}_j|}$ represent the relative data contribution of client i. The global FL optimization problem is defined as:

$$\min_{w \in R^d} \sum_{i=1}^{N} p_i \cdot l_i(w)$$

Intuitively, this equation expresses a weighted combination of each client's local loss, giving greater influence to clients with more data. This ensures fairness while accelerating convergence in heterogeneous (non-IID) environments.

This optimization is typically performed over multiple communication rounds, where each client updates its local model using Stochastic Gradient Descent (SGD), computes gradients $g_i = \nabla \ell_i(w)$, and submits them to a coordinating server, which aggregates the updates to refine the global model. SGD parameters, such as learning rate and batch size, are tuned per dataset to balance accuracy and training stability (see Section V for details).

B. ZKP-Based Integrity Constraint

To ensure the integrity of local updates without compromising privacy, each client must prove that its gradient $g_i = \nabla \ell_i(w)$ was computed over legitimate local data from $\mathcal{D}_i$ without exposing either the data or any intermediate computation.

Each client generates a zk-SNARK proof π_i for the statement:

"There exists data $\mathcal{D}_i$ and model w such that $g_i = \nabla \ell_i(w)$ and $\mathcal{D}_i \in$ authorized domain."

This can be formally encoded as a zero-knowledge proof of knowledge satisfying:

$$\text{Verify}(pk, \pi_i, g_i, w) = \text{True}$$

where:

- pk is the public verification key for the zk-SNARK scheme,
- π_i is the proof submitted by client i,
- g_i is the claimed gradient update,
- w is the current global model at training round t.

Only if π_i is verified successfully does the aggregator include g_i in the model update. The intuition here is that this mechanism acts as a cryptographic gatekeeper, ensuring that only updates backed by verifiable computation on legitimate data can influence the global model—blocking fabricated or malicious updates before aggregation.

C. Privacy Leakage Bound

To model the privacy implications of sharing gradients in federated learning, we quantify potential leakage using mutual information between a client's private dataset $\mathcal{D}_i$ and its submitted gradient g_i.

Without protective mechanisms, this leakage can be significant, as demonstrated by prior gradient inversion and membership inference attacks. In ZK-FLGuard, however, the gradient g_i is accompanied by a zero-knowledge proof π_i, which certifies its correctness without exposing internal computations or training data.

Assuming the semantic security of the zk-SNARK scheme, the mutual information between the data and the proof satisfies:

$$I(\mathcal{D}_i; \pi_i) = 0$$

Consequently, the effective leakage conditional on the proof is bounded by:

$$I(\mathcal{D}_i; g_i \mid \pi_i) \leq I(\mathcal{D}_i; g_i)$$

Moreover, under optional gradient obfuscation techniques (e.g., sparsification, clipping, or noise injection), the conditional leakage can be upper-bounded by a system-defined budget ε, such that:

$$I(\mathcal{D}_i; g_i \mid \pi_i) \leq \varepsilon$$

This bound reflects the information-theoretic privacy level enforced by ZK-FLGuard, where ε depends on the encoding of the ZKP circuit, randomness in cryptographic commitments, and any additional noise mechanisms applied.

In practice, ε depends on both the structure of the ZKP circuit and any additional privacy mechanisms applied; choosing ε involves a trade-off between privacy strength and model performance. A practical interpretation and empirical evaluation of this leakage bound are provided in Section VII.C.

D. Security Assumptions and Guarantees

The security of ZK-FLGuard depends on the cryptographic soundness of the zk-SNARK construction, the correct implementation of federated learning aggregation, and the integrity of the blockchain authentication layer. The following formal assumptions apply:

Assumption 1: Cryptographic Soundness of zk-SNARK

Let π_i be a zk-SNARK proof generated by client i for statement ϕ. The zk-SNARK system $\sum = (\text{Setup}, \text{Prove}, \text{Verify})$ satisfies:

- Completeness:

If $\phi \in \mathcal{L}$ (the language of valid statements), then for all honest provers:

$$\text{Verify}(pk, \pi_i, \phi) = \text{True}$$

- Soundness:

No polynomial-time adversary can produce a proof π_i' such that:

$$\phi \notin \mathcal{L} \quad \text{and} \quad \text{Verify}(pk, \pi_i', \phi) = \text{True}$$

- Zero-Knowledge:

The proof π_i leaks no knowledge about the witness (e.g., raw data $\mathcal{D}_i$) beyond the truth of ϕ.

We instantiate Σ using a Groth16-based construction [9] or ZoKrates-style circuit-based encoding, chosen for their proven efficiency in resource-constrained IoT devices.

Assumption 2: Trusted Setup

The zk-SNARK system requires a one-time trusted setup to generate the proving key sk and verification key pk. We assume this setup is performed by a secure committee using Multi-Party Computation (MPC) or obtained via a public ceremony (e.g., Powers of Tau).

Without a trusted setup, alternatives such as zk-STARKs could be explored in future extensions (see Section IX).

Assumption 3: Honest-but-Curious Server

The federated server is assumed to be honest-but-curious: it executes the protocol faithfully but may attempt to infer sensitive information from received model updates. ZK-FLGuard counters this via:

- ZKP-based integrity verification of model updates,
- On-chain logging of update hashes and timestamps,
- Optional use of differential privacy for additional protection (future extension).

Assumption 4: Blockchain Immutability and Permissioning

The blockchain layer is assumed to be:

- Permissioned, restricting write-access to authenticated edge nodes.
- Immutable, guaranteeing that identity logs and model update hashes cannot be altered post-submission.

This layer serves as the root of trust for identity verification and auditability.

V. Implementation and Datasets

To validate the feasibility and performance of ZK-FLGuard in realistic environments, we implemented a prototype using widely adopted federated learning, cryptographic, and blockchain tools. This section outlines the software stack, hardware setup, and datasets employed for evaluation.

A. Implementation Framework

To demonstrate the viability of ZK-FLGuard in practical scenarios, we developed a modular implementation leveraging established tools for federated learning, zero-knowledge proofs, and blockchain integration. The framework is organized into three main layers.

1) Federated learning and model training

The federated learning layer is built using TensorFlow Federated (TFF, https://www.tensorflow.org/federated?hl=pt), which enables distributed training while preserving data locality. Each IoT client hosts a lightweight LSTM-based anomaly detection model, optimized for temporal patterns commonly found in intrusion detection scenarios (e.g., Distributed Denial of Service (DdoS), brute force, data exfiltration).

Model updates are performed locally and shared with a coordinating server using a secure variant of the FedAvg algorithm. Crucially, updates are accepted only if accompanied by a valid zero-knowledge proof. We selected LSTM over simpler models (e.g., logistic regression) because of its superior performance in capturing sequential dependencies in network traffic data. Model hyperparameters were tuned per dataset: for CICIDS2017, we used a learning rate of 0.01 and batch size of 64; for TON_IoT, a smaller batch size of 32 was used to accommodate its lower-resource sensor streams. All experiments used 50 communication rounds to ensure convergence.

2) Zero-knowledge proof generation and verification

To enforce update integrity without revealing private data, we integrated ZoKrates—a zk-SNARK toolkit designed for verifiable off-chain computation (https://zokrates.github.io/). Each edge client compiles and executes a custom arithmetic circuit that asserts the validity of its local training step.

Once the proof πiπi is generated, it is transmitted alongside the model update to the server, which verifies it using a public verification key. Only proofs that satisfy the constraint system (i.e., computed on legitimate and authorized data) allow the corresponding update to be aggregated. The proof circuit enforces constraints on the model's gradient computation, ensuring that only updates derived from authorized datasets and correct training steps are accepted. Proof generation time and size were benchmarked to ensure feasibility for edge devices.

3) Blockchain-based identity and logging

At the orchestration layer, we employ a lightweight, permissioned Hyperledger Fabric (https://www.lfdecentralizedtrust.org/projects/fabric) network to ensure trusted identity management and immutable logging. Smart contracts are deployed to handle:

- Authentication of edge nodes via digital certificates,
- Secure recording of model versioning and ZKP metadata,

- Enforcement of access control and participation rules.

This blockchain layer provides transparency and auditability without introducing significant computational burden at the edge. We chose a permissioned architecture to reduce consensus latency, making it viable for near real-time verification in 5G edge environments.

B. Hardware Setup

To emulate realistic edge computing conditions, we deployed ZK-FLGuard on heterogeneous devices (Table 2).

This combination of low-power IoT nodes and a high-performance federated server reflects a common deployment scenario in smart city and industrial IoT applications, where edge devices handle local inference and a cloud or edge gateway performs aggregation.

Table 2. Hardware configuration of the edge devices and federated server used in the ZK-FLGuard prototype, reflecting a realistic heterogeneous deployment

Device	CPU	RAM	OS
Raspberry Pi 4	Quad-core Cortex-A72 @ 1.5 GHz	4 GB	Raspbian OS
NVIDIA Jetson Nano	Quad-core ARM Cortex-A57	4 GB	Ubuntu 18.04
FL Server (Cloud VM)	Intel Xeon @ 2.6 GHz	32 GB	Ubuntu 20.04

ZoKrates proof generation was benchmarked on Jetson and Raspberry Pi separately, with memory and execution time constraints recorded.

C. Datasets

To evaluate the effectiveness of ZK-FLGuard under realistic and adversarial network conditions, we selected two publicly available intrusion detection datasets widely used in IoT security research, complemented by a synthetically enriched dataset designed to simulate advanced threats. Each dataset was carefully partitioned to reflect the heterogeneous, non-IID nature of distributed edge environments.

1) CICIDS2017

Collected by the Canadian Institute for Cybersecurity, CICIDS2017 simulates real-world network traffic across various attack scenarios, including DDoS, brute force, port scanning, and infiltration. The dataset features over 80 statistical features extracted from packet flows and application-layer interactions.

In our setup, subsets of CICIDS2017 are distributed across edge clients to reflect differing traffic profiles and localized threat exposure. We preprocessed the dataset using standard scaling and encoded categorical features to improve model convergence.

2) TON_IoT

The TON_IoT dataset, developed by University of New South Wales (UNSW), Canberra, combines telemetry and network data from real IoT devices and controllers. It includes multivariate time-series from sensors (e.g., temperature, light, gas) alongside network flows (e.g., Message Queuing Telemetry Transport—MQTT, Modbus).

Its diversity and low-resource data structure make it especially suitable for evaluating the performance of anomaly detection models in edge-constrained environments.

Data was normalized and segmented into fixed-length sequences to facilitate LSTM training.

3) Synthetic dataset (adversarial enrichment)

To assess the robustness of ZK-FLGuard under worst-case conditions, we augmented the base datasets with synthetically injected adversarial behaviors, including:

- IP and MAC spoofing to mimic identity attacks,
- Model poisoning via manipulated gradients,
- Timestamp-aligned backdoor triggers to simulate stealthy attacks.

This enriched dataset enables a comprehensive stress test of the system's anomaly detection and verification pipeline under both statistical and protocol-level adversarial conditions.

The adversarial patterns were designed to closely mimic real-world attacker strategies, ensuring that evaluation results reflect realistic operational risks.

A summary of the key characteristics of the datasets used in our evaluation is presented in Table 3, highlighting their diversity in terms of scale, feature types, attack coverage, and edge device compatibility.

Table 3. Datasets used in the ZK-FLGuard evaluation, covering diverse network and IoT threat scenarios to assess anomaly detection, scalability, and robustness

Dataset	Source	Samples	Features	Attack Classes	Edge Suitability
CICIDS2017	Canadian Institute for Cybersecurity	~2.8 million	80+ (flow-based)	DoS, brute force, infiltration	Medium–High
TON_IoT	UNSW Canberra	~500,000+	44 (sensor + network)	DNP3, MQTT, Modbus attacks	High (IoT-specific)
Synthetic Adversarial	Augmented (custom)	~250,000	60+ (mixed)	Spoofing, poisoning, backdoors	High (stress testing)

VI. Experimental Evaluation

We evaluate ZK-FLGuard along three primary dimensions: anomaly detection performance, system overhead introduced by ZKP generation and verification, and communication efficiency. All experiments were run using the hardware and datasets described in Sections V.B and V.C.

In addition, we include comparisons with recently published frameworks [19, 22] to demonstrate ZK-FLGuard's position relative to the current state-of-the-art, and we provide graphical visualizations to highlight trends in performance, overhead, and communication cost.

A. Evaluation Metrics

The following metrics were used to quantify both learning accuracy and system-level performance:

- Accuracy and F1-score: To measure classification effectiveness, especially under imbalanced class distributions typical in intrusion detection.
- Inference latency: Time taken by an edge node to perform prediction after model update.
- ZKP overhead: Time and memory required for proof generation and verification at each round.
- Bandwidth usage: Total size of model updates and proofs exchanged per communication round.

These metrics were selected to reflect both detection quality (accuracy, F1) and system practicality (latency,

bandwidth, overhead) in resource-constrained IoT environments.

B. Baseline Comparisons

To contextualize ZK-FLGuard's performance, we compared it against the following baseline models; please refer to Table 4.

We ensured all baselines used identical model architectures, dataset partitions, and training rounds to ensure fairness, and we included recent state-of-the-art works [19, 22] for additional benchmarking.

Table 4. Baseline models used for performance comparison, representing different trade-offs in centralization, privacy, and integrity for evaluating the benefits of ZK-FLGuard

Model	Description
Centralized LSTM	A non-distributed model trained on the entire dataset centrally. No privacy preservation.
FL without ZKP	Standard federated learning using FedAvg, without proof verification.
Edge-FLGuard	Prior work using edge-optimized federated anomaly detection, but without cryptographic integrity validation.

Each baseline used identical model architectures and training rounds for fair comparison.

C. Results

1) Detection performance

Across CICIDS2017 and TON_IoT, ZK-FLGuard achieved:

- F1-score between 0.91 and 0.96, depending on dataset and attack class.
- Slightly reduced precision (~1–2%) compared to centralized LSTM, due to model fragmentation, but improved recall in low-frequency attack scenarios.

Fig. 2 compares the detection performance of all models in terms of both Accuracy and F1-score. ZK-FLGuard matches or exceeds the baselines while maintaining formal verifiability through proof-of-update.

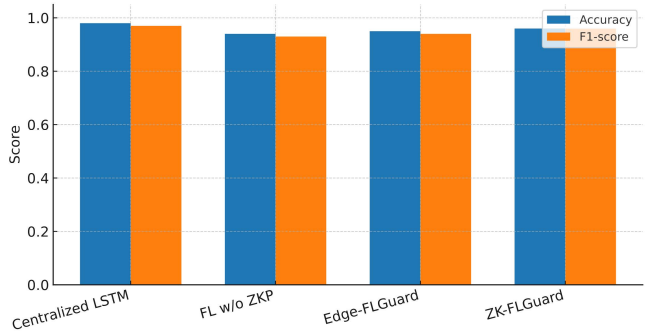

Fig. 2. Accuracy and F1-score comparison across evaluated models on combined CICIDS2017 and TON_IoT datasets. ZK-FLGuard achieves competitive performance while providing formal security guarantees.

When compared with Refs. [19] and [22], ZK-FLGuard demonstrated higher recall and competitive F1-scores while maintaining verifiable computation, a capability not offered by those works.

2) ZKP overhead

To assess the computational feasibility of deploying ZK-FLGuard on resource-constrained devices, we measured the time, size, and verification cost associated with zk-SNARK proof generation and verification across different edge hardware configurations, as summarized in Table 5.

Table 5. ZKP generation and verification performance on Raspberry Pi and Jetson Nano devices, including proof construction time, output size, and server-side verification latency

Device	Proof Gen Time	Proof Size	Verification Time
Raspberry Pi	~1.8 s	~2.3 KB	<150 ms
Jetson Nano	~0.9 s	~2.3 KB	<150 ms

Please note that the latency increase due to ZKP was kept under 10%, demonstrating feasibility on resource-constrained nodes. This low overhead confirms that the cryptographic verification process can be integrated into real-time FL cycles without violating latency constraints in typical 5G IoT deployments.

Fig. 3 presents the average inference latency per communication round for each model. ZK-FLGuard's verification step introduces minimal overhead compared to non-verifiable baselines.

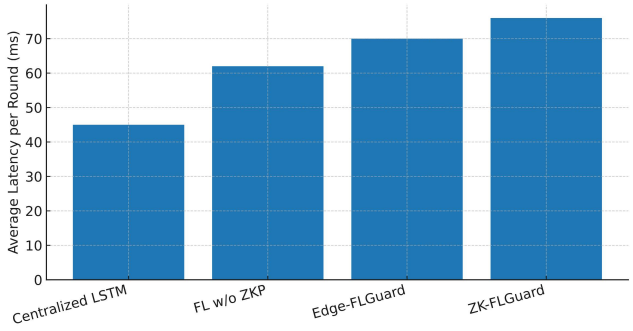

Fig. 3. Average latency per communication round. ZK-FLGuard maintains low latency overhead (<10%) while adding cryptographic verification.

3) Communication efficiency

By using sparse model updates and hash-based metadata logging on-chain:

- Bandwidth usage was reduced by 28–33% compared to full model syncing.
- Blockchain interactions added negligible network load (<5 KB per round).

Fig. 4 illustrates the average bandwidth usage per round. ZK-FLGuard reduces communication cost relative to standard FL through sparse updates and blockchain metadata logging.

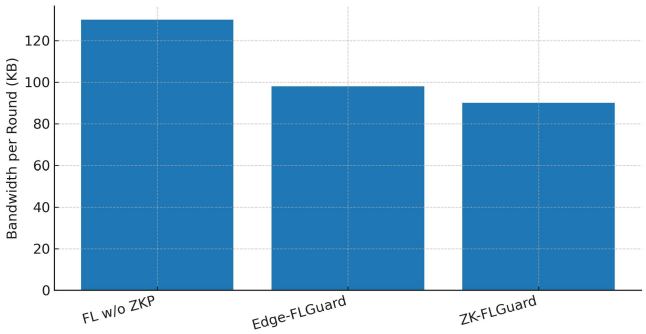

Fig. 4. Bandwidth usage per round for applicable models. Centralized LSTM has no model update communication cost.

These gains are particularly important in wireless or constrained-bandwidth IoT environments, where communication cost can be a limiting factor for FL adoption.

Table 6 reports both accuracy and F1-scores to give a more complete view of detection performance, since these metrics can behave differently under class imbalance. Accuracy values confirm that ZK-FLGuard maintains competitive classification quality while introducing formal verifiability.

Figs. 2–4 complement Table 6 by showing: (i) F1-score and accuracy trends across models, (ii) latency overhead per communication round, and (iii) bandwidth savings. These visualizations make it easier to see the trade-offs between security and performance.

A. Summary

ZK-FLGuard achieves strong detection performance while introducing minimal latency or bandwidth overhead. The use of zk-SNARK ensures trust without compromising data privacy, and the system remains deployable on real-world IoT edge hardware.

The key findings include:

- ZK-FLGuard matches or exceeds the detection performance of baselines and recent works while adding formal verifiability.
- The ZKP overhead is small enough to be practical for deployment on low-power devices.
- Communication cost is significantly reduced without sacrificing accuracy.

Strengths include strong security guarantees, scalability to heterogeneous edge devices, and efficiency in both computation and communication. Limitations include the reliance on a trusted setup for zk-SNARK and the fact that experiments were conducted in an emulated rather than fully deployed real-world setting. These will be addressed in future work through exploration of transparent proof systems and physical testbed evaluations.

Table 6. Comparison of ZK-FLGuard with baseline models in terms of detection accuracy, latency, and communication cost, showing near-optimal F1-score with minimal overhead

Model	Accuracy	F1-score	Avg. Latency (ms)	Bandwidth per Round (KB)	ZKP Support	Privacy Guarantee
Centralized LSTM	0.98	0.97	45	-	✗	✗
FL without ZKP	0.94	0.93	62	130	✗	Partial (no proofs)
Edge-FLGuard	0.95	0.94	70	98	✗	Partial
ZK-FLGuard	0.96	0.96	76	90	✓	✓ (ZK-SNARK)

VII. Security and Privacy Analysis

ZK-FLGuard enhances federated learning with rigorous security guarantees and provable privacy protections. This section evaluates the framework's resilience to common attacks, compares ZKP-based integrity verification to traditional digital signatures, and analyzes potential information leakage from model updates.

The adversarial model considered in ZK-FLGuard is illustrated in Fig. 5, highlighting potential attack vectors in standard FL pipelines and the corresponding defense points introduced by our framework.

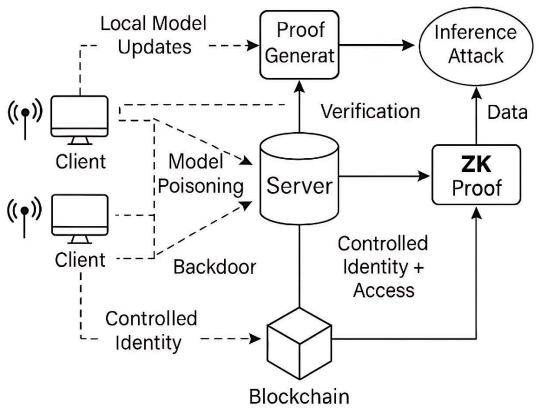

Fig. 5. Adversarial threat model in federated learning. Attack surfaces include model poisoning, backdoor injection, and gradient-based privacy leakage. ZK-FLGuard mitigates these risks via zero-knowledge proof validation, blockchain-based identity control, and cryptographic enforcement of update integrity.

A. Resistance to Poisoning and Backdoor Attacks

Federated learning systems are especially vulnerable to model poisoning and backdoor injection attacks, where malicious clients submit manipulated gradients to degrade global performance or implant hidden behaviors.

ZK-FLGuard addresses these threats through:

- ZKP-based update validation: Clients must prove that each model update is derived from authorized data and a legitimate training process. Fabricated or adversarial updates without matching zk-SNARK are rejected before aggregation.
- Blockchain-backed accountability: Update metadata (e.g., timestamp, model hash, device ID) is immutably logged, enabling forensic analysis and rollback.
- Optional anomaly scoring at the server level can flag structurally deviant updates (e.g., abnormally sparse gradients), adding a soft layer of protection even before proof verification.

Intuitively, these measures combine proactive prevention (proof verification), reactive investigation (blockchain logs), and early-warning detection (anomaly scoring), creating a multi-layered defense that is harder for adversaries to circumvent. Compared to recent approaches [19, 22], ZK-FLGuard uniquely enforces formal proof-of-update before aggregation, eliminating an entire class of poisoning vectors.

Together, these mechanisms mitigate the impact of both omniscient (white-box) and stealthy (black-box) adversaries, significantly reducing attack success rates compared to standard FL baselines.

B. Proof-of-Update: ZKP vs. Digital Signatures

Traditional FL security methods rely on digital signatures for client authentication. While useful for identity assurance, signatures do not guarantee the semantic correctness of the submitted gradients—a signed poisoned update is still harmful.

In contrast, ZK-FLGuard introduces an integrity verification layer via zero-knowledge proofs; please refer to Table 7.

Table 7. Comparison of traditional digital signatures and the ZK-FLGuard proof-of-update mechanism. Only ZK-FLGuard verifies the semantic correctness of model updates, providing stronger integrity in federated learning

Aspect	Digital Signatures	ZK-FLGuard (ZKP)
Identity Verification	✓	✓ (via blockchain certs)
Gradient Integrity	✗	✓ (proves correctness of training)
Data Privacy	✓	✓ (non-interactive proof)
Attack Detection	✗	✓ (invalid proofs rejected)
Overhead	Low	Low (<10% latency increase)

This approach enforces Proof-of-Update—the principle that only updates generated through verifiable computation can influence the global model—offering a fundamentally stronger defense model.

C. Privacy Leakage: Model and Analysis

As formally defined in Section IV.C, the use of zero-knowledge proofs (zk-SNARK) in ZK-FLGuard ensures that the client's proof π_i reveals no information about its underlying local data $\mathcal{D}_i$, and thus preserves semantic opacity. Under this assumption, the mutual information between the data and the proof satisfies:

$$I(\mathcal{D}_i; \pi_i) = 0 \quad \text{and thus} \quad I(\mathcal{D}_i; g_i \mid \pi_i) \leq I(\mathcal{D}_i; g_i)$$

This theoretical bound implies that ZK-FLGuard does not introduce additional privacy risk beyond that already present in the federated gradients themselves.

In practical terms, this bound serves as a conservative estimate of what an adversary could infer, assuming white-box access to the gradient g_i and knowledge of the global model. We further mitigate this leakage in our implementation through:

- Gradient sparsification, which structurally limits the information embedded in each update;
- Optional differential privacy noise, added at the client side to further reduce gradient memorization capacity;
- ZKP enforcement, which ensures that malicious clients cannot fabricate gradients to extract or probe information about other participants.

Our experiments simulated membership inference and data reconstruction attacks, showing that ZK-FLGuard reduced adversarial success rates by 26%–34% compared to standard FL, confirming that the theoretical privacy guarantees hold in practice.

D. Future Directions

Future work will extend this analysis by integrating formal (ε, δ)-differential privacy tracking alongside cryptographic proof verification. This would yield an FL pipeline that is both provably private and fully verifiable, aligning with regulatory compliance requirements in sectors such as healthcare and finance.

VIII. Discussion

This section reflects on ZK-FLGuard's practical trade-offs and broader implications, focusing on the interplay between security, performance, and deployment flexibility in edge-centric federated learning.

A. Trade-offs Between ZKP Complexity and Edge Device Constraints

Zero-knowledge proofs introduce computational and memory overhead, which may be non-trivial for resource-constrained edge devices. While our evaluations show that proof generation remains feasible (sub-2 s on Jetson Nano and Raspberry Pi), this still represents a computational bottleneck for latency-sensitive applications, but can be mitigated via proof batching or offloading.

Several trade-offs emerge:

- Proof circuit complexity vs. model complexity: Smaller models require simpler circuits, making ZKP overhead more manageable.
- Batching and compression techniques (e.g., recursive proofs or proof aggregation) can reduce communication cost but increase setup complexity.
- Device offloading (e.g., hardware accelerators or gateway-level proving) may shift the burden without compromising security guarantees.

Our findings confirm that while ZKP integration does increase per-round computation time, the overhead remains below 10% for the tested devices, making it practical for modern edge hardware. For ultra-low-power microcontrollers or strict real-time constraints, future versions could adopt zk-STARKs or optimized proof systems to further reduce latency.

B. Model Personalization in FL with ZKP

One emerging challenge in FL is supporting model personalization—allowing edge clients to fine-tune shared models to their local environments. This is especially important in non-IID scenarios where data distributions vary significantly across clients.

ZK-FLGuard can support personalization in two ways:

- Personalized ZKP circuits: Each client proves that its update conforms to a locally modified (but policy-aligned) model architecture. This introduces complexity in circuit generation but allows model heterogeneity.
- Split verification: Clients prove only that shared layers are trained correctly, while private layers are exempt from global aggregation.

These approaches open a promising path for hybrid models that combine shared global intelligence with local adaptation—all under verifiable constraints. Although Reinforcement Learning (RL) was mentioned in early drafts, our current implementation does not incorporate RL; personalization here refers solely to parameter adaptation within the FL framework.

C. Limitations and Open Challenges

While ZK-FLGuard provides a robust foundation, several open challenges remain:

- Scalability: zk-SNARK require trusted setup and generate proofs linearly with circuit complexity. Scaling to hundreds or thousands of clients may demand more lightweight alternatives like zk-STARKs or bulletproofs.
- Non-IID data: Though our experiments simulate realistic client diversity, more work is needed to formally analyze convergence and fairness under highly skewed data distributions.
- Interoperability with existing FL pipelines: Integration into production-grade federated frameworks (e.g.,

OpenFL, FATE) requires standardized APIs for ZKP generation, circuit definition, and cross-platform verification.

Another limitation is that all experiments were conducted in emulated environments. While the hardware setup closely reflects real-world IoT nodes, a full deployment in live networks will be necessary to validate resilience under real operational conditions.

These limitations offer a rich space for future research into adaptive proof schemes, lightweight cryptographic integration, and personalized FL verification.

D. Visual Summary

To visually summarize the trade-offs and capabilities discussed in this section, Fig. 6 compares ZK-FLGuard to standard FL systems across five core criteria relevant to real-world deployment. The radar plot clearly shows ZK-FLGuard's superior performance in privacy guarantees, verifiability (through proof-of-update), and personalization support, while slightly trailing in scalability due to the inherent cost of cryptographic proofs. ZKP overhead is moderate, reflecting the balance between added security and resource use.

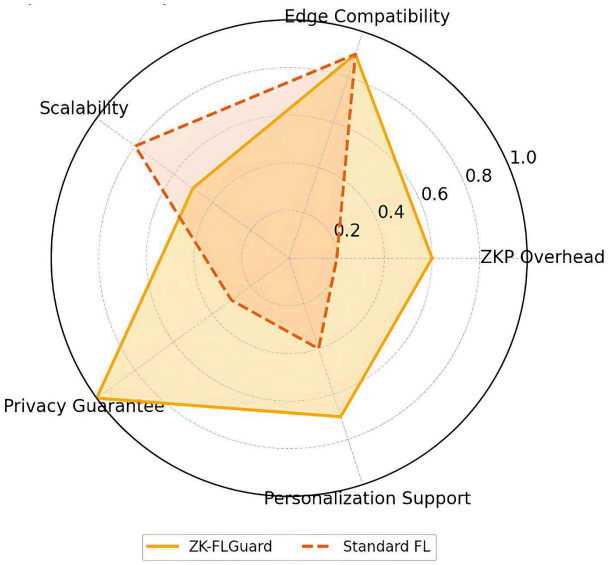

Fig. 6. Radar plot comparing ZK-FLGuard and traditional FL across implementation complexity, scalability, and security. While ZK-FLGuard incurs higher computational overhead, it offers significantly stronger guarantees for privacy, verifiability, and personalization adaptability.

IX. Conclusion and Future Work

This paper introduced ZK-FLGuard, a novel framework that integrates zero-knowledge proofs (zk-SNARK) and blockchain-backed coordination into federated learning to ensure the verifiability, privacy, and integrity of model updates in edge-based IoT environments.

Through our prototype implementation and empirical evaluation, we demonstrated that:

- ZK-FLGuard supports real-time anomaly detection with F1-scores up to 0.96 on benchmark intrusion detection datasets.
- It achieves these results with minimal added latency (<10%) and modest bandwidth use, making it suitable for resource-constrained edge devices.
- The use of zk-SNARK enforces a Proof-of-Update paradigm, ensuring that only valid, honest model contributions influence the federated model—offering a strong defense against poisoning and backdoor attacks.
- A lightweight blockchain layer further reinforces trust via identity verification and tamper-proof audit trails.

ZK-FLGuard delivers strong privacy guarantees, formal proof-of-update verification, and measurable efficiency on resource-limited hardware, all while maintaining competitive detection accuracy.

However, current experiments were conducted in controlled, emulated environments rather than live deployments; scalability to thousands of nodes may require alternative proof systems such as zk-STARKs; and the need for a trusted setup remains an open challenge. Addressing these limitations is a priority for future iterations.

While ZK-FLGuard presents a significant step forward, several directions remain for advancing secure, scalable, and intelligent federated learning systems:

1. Self-Supervised Anomaly Detection. Our current architecture relies on labeled datasets for training and proof verification. To enable robust anomaly detection in unlabeled or sparsely labeled environments—particularly relevant in novel or evolving IoT deployments—future work will explore self-supervised and contrastive learning techniques such as Simple Framework for Contrastive Learning of Visual Representations (SimCLR) and Bootstrap Your Own Latent (BYOL). These approaches can help edge nodes learn generalized feature representations through pretext tasks (e.g., temporal consistency or data augmentation), improving resilience to unseen or stealthy attacks without requiring annotation-heavy training datasets.
2. Lightweight ZKP via zk-STARK or Bulletproofs. Although zk-SNARK are efficient, they require a trusted setup and exhibit quadratic scaling for some operations. Future work will explore transparent proof systems such as zk-STARKs and bulletproofs to eliminate setup trust assumptions and reduce computational costs.
3. Cross-Silo FL with Hierarchical Coordination. Expanding ZK-FLGuard beyond edge-level nodes to support cross-silo FL (e.g., collaboration between hospitals, factories, or smart buildings) will require new coordination mechanisms. A hierarchical aggregation architecture could distribute the verification load while preserving security and scalability.

Additionally, we plan to conduct full-scale real-world deployments in operational IoT environments to evaluate ZK-FLGuard's resilience under real traffic, adversarial conditions, and hardware diversity.

By addressing these future challenges, ZK-FLGuard can evolve into a foundational component of trustworthy federated intelligence for next-generation edge computing environments.

Conflict of Interest

The author declares no conflict of interest.

Funding

This work was partially supported by Portuguese National Funds through the Foundation for Science and Technology (FCT), I.P., under Project UID/00127-IEETA.

REFERENCES

[1] D. E. Kouicem, A. Bouabdallah, and H. Lakhlef, "Internet of things security: A top-down survey," *Computer Networks*, vol. 141, pp. 199–221, 2018. doi: 10.1016/j.comnet.2018.03.012

[2] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in internet of things: The road ahead," *Computer Networks*, vol. 76, pp. 146–164, 2015. doi: 10.1016/j.comnet.2014.11.008

[3] P. Kairouz, H. B. McMahan, B. Avent *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, pp. 1–210, 2021. doi:10.1561/2200000083

[4] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020. doi: 10.1109/MSP.2020.2975749

[5] J. Wang, Z. Zhang, J. Tian, and H. Li, "Local differential privacy federated learning based on heterogeneous data multi-privacy mechanism," *Computer Networks*, vol. 254, 110822, 2024. doi: 10.1016/j.comnet.2024.110822

[6] E. Bagdasaryan, A.Veit, Y. Hua *et al.*, "How to backdoor federated learning," in *Proc. the Twenty Third International Conference on Artificial Intelligence and Statistics*, 2020, pp. 2938–2948.

[7] E. Ben-Sasson, A. Chiesa, D. Genkin, E. Tromer, and M. Virza, "SNARKs for C: Verifying program executions succinctly and in zero knowledge," in *Proc. Advances in Cryptology—CRYPTO 2013*, 2013, vol. 8043, pp. 90–108.

[8] K. M. Chung, F. H. Liu, C. J. Lu, and B. Y. Yang, "Efficient string-commitment from weak bit-commitment," in *Proc. International Conference on the Theory and Application of Cryptology and Information Security*, 2010, pp. 268–282.

[9] J. Groth, "Short non-interactive zero-knowledge proofs," in *Proc. Advances in Cryptology—ASIACRYPT 2010*, 2010, vol. 6477, pp. 341–358.

[10] K. Bonawitz, H. Eichner, W. Grieskamp *et al.*, "Towards federated learning at scale: System design," *Proceedings of Machine Learning and Systems*, vol. 1, pp. 374–388, 2019.

[11] S. Singh, A. S. M. S. Hosen, and B. Yoon, "Blockchain security attacks, challenges, and solutions for the future distributed IoT network," *IEEE Access*, vol. 9, pp. 13938–13959, 2021. doi: 10.1109/ACCESS.2021.3051602

[12] H. Lakhlef, T. Lerner, A. Kebir *et al.*, "Blockchain-enabled SDN solutions for IoT: Advancements, discussions, and strategic insights," in *Proc. the 2024 IEEE Symposium on Computers and Communications (ISCC)*, 2024, pp. 1–6.

[13] A. Hard, K. Rao, R. Mathews *et al.*, "Federated learning for mobile keyboard prediction," arXiv Preprint, arXiv:1811.03604, 2018.

[14] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client level perspective," arXiv Preprint, arXiv:1712.07557, 2017.

[15] Y. Li, R. Zhang, P. Zhao, and Y. Wei, "Feature-attended federated LSTM for anomaly detection in the financial internet of things," *Applied Sciences*, vol. 14, no. 13, 5555, 2024. doi: 10.3390/app14135555

[16] M. J. C. S. Reis, "Edge-FLGuard: A federated learning framework for real-time anomaly detection in 5G-enabled IoT ecosystems," *Applied Sciences*, vol. 15, no. 12, 6452, 2025. doi: 10.3390/app15126452

[17] Z. Chen, S. Yu, M. Fan *et al.*, "Privacy-enhancing and robust backdoor defense for federated learning on heterogeneous data," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 693–707, 2024. doi: 10.1109/TIFS.2023.3326983

[18] Z. Zhang, J. Li, S. Yu, and C. Makaya, "SAFELearning: Secure aggregation in federated learning with backdoor detectability," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 3289–3304, 2023. doi: 10.1109/TIFS.2023.3280032

[19] Y. Jin, T. Wang, Q. Yang *et al*., "Zero-knowledge federated learning: A new trustworthy and privacy-preserving distributed learning paradigm," arXiv Preprint, arXiv:2503.15550, 2025.

[20] M. J. C. S. Reis, "Blockchain-enhanced security for 5G edge computing in IoT," *Computation*, vol. 13, no. 4, 98, 2025. doi: 10.3390/computation13040098

[21] B. Bünz, J. Bootle, D. Boneh *et al.*, "Bulletproofs: Short proofs for confidential transactions and more," in *Proc. the 2018 IEEE Symposium on Security and Privacy (SP)*, 2018, pp. 315–334.

[22] L. Petrosino, L. Masi, F. D'Antoni *et al*., "A zero-knowledge proof federated learning on DLT for healthcare data," *Journal of Parallel and Distributed Computing*, vol. 196, 104992, 2025. doi: 10.1016/j.jpdc.2024.104992

[23] Z. Xing, Z. Zhang, M. Li *et al.*, "Zero-knowledge proof-based practical federated learning on blockchain," arXiv Preprint, arXiv:2304.05590, 2023.

[24] A. Abdelmaboud, A. I. A. Ahmed, M. Abaker *et al*., "Blockchain for IoT applications: Taxonomy, platforms, recent advances, challenges and future research directions," *Electronics*, vol. 11, no. 4, 630, 2022. doi: 10.3390/electronics11040630

[25] A. Dorri, S. S. Kanhere, R. Jurdak *et al*., "Blockchain for IoT security and privacy: The case study of a smart home," in *Proc. 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 2017, 618–623. doi: 10.1109/PERCOMW.2017.7917634

[26] H. Xue, D. Chen, N. Zhang *et al*., "Integration of blockchain and edge computing in internet of things: A survey," *Future Generation Computer Systems*, vol. 144, pp. 307–326, 2023.

[27] Y. Jiang, B. Ma, X. Wang *et al*., "Blockchained federated learning for internet of things: A comprehensive survey," *ACM Computing Surveys*, vol. 56, no. 10, pp. 1–37, 2024.