Journal of Theoretical and Applied Information Technology
15th May 2026. Vol.104. No.9

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AUTHENTICATING IOT DEVICES WITHOUT REVEALING THEIR RF FINGERPRINTS: A ZERO-KNOWLEDGE FRAMEWORK ON BLOCKCHAIN

YASSINE LKHALIDI[1], MOHAMED LKHALIDI [2], HATIM KHARRAZ AROUSSI[1], ACHRAF TIFERNINE [1]

[1] University of Ibn Tofail, Department of Computer Science, Kenitra, Morocco
[2] Moulay Ismail University, Institute of Advanced Education, Meknes, Morocco

E-mail: [1]lkhalidi.yassine@uit.ac.ma, [1]hatim.kharrazaroussi@uit.ac.ma, [1]achraf.tifernine@uit.ac.ma,[2] m.lkhalidi@edu.umi.ac.ma

ABSTRACT

IoT device authentication must resist impersonation and credential theft while respecting the computational constraints of edge devices. Existing frameworks rely on static cryptographic keys that, once extracted, enable full impersonation, whereas RF fingerprinting schemes that bind identity to hardware imperfections transmit and store device templates in plaintext, exposing them to template theft and linkability attacks. ZK-RFAuth is a three-phase authentication framework that integrates Siamese neural network-based RF fingerprinting, Groth16 zero-knowledge proof (ZKP) verification, and proof-of-authority blockchain logging. During registration, a Siamese convolutional network extracts a compact embedding from raw I/Q samples and commits a Poseidon hash of the quantized mean template on-chain. During verification, the prover generates a Groth16 proof demonstrating that the L1 distance between a fresh embedding and the registered template falls below a per-device threshold without revealing either vector. The proof and authentication outcome are recorded on-chain for tamper-evident auditing. Evaluated on the WiSig dataset (28 WiFi transmitters, 224,000 frames), ZK-RFAuth achieves 91.4% closed-set accuracy and 2.25% equal error rate at embedding dimension d = 64, with 88.4% genuine acceptance rate and 70.8% open-set rogue rejection using per-device P95 thresholds. The ZKP circuit requires only 972 rank-1 constraint system (R1CS) constraints over 100× fewer than an equivalent SHA-256 circuit producing 144-byte proofs verifiable in approximately 3 ms.
Keywords: *IoT authentication, Zero-Knowledge Proofs, RF fingerprinting, Siamese network, Blockchain.*

1. INTRODUCTION

The deployment of Internet of Things (IoT) devices has reached a scale at which authentication failures carry systemic risk. The global installed base of connected IoT devices is projected to exceed 25 billion by 2030 [1], spanning healthcare, industrial control, smart-grid metering, and vehicular communications. Across these domains, devices collect sensitive measurements, actuate physical processes, and relay safe-ty-critical commands, contexts in which device impersonation or unauthorized access can cause harm well beyond the digital domain. Designing authentication mechanisms that scale to this population while respecting the severe memory and compute constraints of IoT hardware is therefore one of the defining challenges of current network security research. Prevailing authentication practice binds device identity to a cryptographic secret stored in memory: an Elliptic Curve Cryptography (ECC) private key, an X.509 certificate, or a pre-shared symmetric key. These mechanisms are computationally efficient and formally well-understood, but they share a structural weakness: identity is bound to a value in non-volatile storage, not to the physical hardware executing the code. An adversary who extracts the private key through a side-channel attack, firmware dump, or supply-chain compromise gains an identity token that is indistinguishable from the legitimate device, with no mechanism to detect the breach until fraudulent activity is observed [2]. Hardware-based countermeasures such as Physically Unclonable Functions (PUFs) embed identity in manufacturing variability and resist key extraction, but they require dedicated silicon circuits absent from commodity

IoT modules the ESP32-S3, ADALM-PLUTO, or Raspberry Pi Zero that constitute the majority of deployed IoT nodes.

Radio Frequency (RF) fingerprinting offers a middle path between these extremes. Each radio transmitter emits involuntary imperfections arising from its manufacturing tolerances: IQ imbalance, carrier frequency offset (CFO), phase noise, and digital-to-analog converter (DAC) nonlinearities. These impairments are stable across repeated transmissions, unique to each physical unit, and extractable passively by any software-defined radio (SDR) receiver, no additional hardware on the device is required. Recent work has applied deep convolutional neural networks (CNNs) to RF fingerprinting at scale, demonstrating closed-set identification accuracy exceeding 99% across 10,000 LoRa transmitters [3].

Siamese network architectures have pushed the frontier further: by learning a metric space rather than a fixed classification boundary, they enable one-shot device enrollment without retraining, a property essential for IoT networks where devices register and deregister continuously [4,5]. Deploying RF fingerprinting in a production authentication system, however, immediately exposes a critical privacy vulnerability. Authentication requires comparing a device's current RF emission against a stored template, an embedding vector of floating-point values computed by a neural network. If that template is stored in plaintext at the authentication server, a compromised server gives an attacker the means to profile every enrolled device's RF characteristic permanently. This threat is structurally identical to storing biometric templates without cryptographic protection: once exposed, the compromise cannot be revoked, because a device's hardware imperfections are intrinsic and immutable. Zero-knowledge proofs (ZKPs) address precisely this class of problem. A ZKP allows one party (the device, acting as prover) to convince another party (the authentication server, acting as verifier) that a private statement is true here, that its current RF embedding matches a registered template within a distance threshold θ without disclosing the embedding itself.

The three research threads most relevant to this work deep learning-based RF fingerprinting, ZKP-based authentication, and lightweight IoT authentication frameworks have each progressed substantially but have never been integrated into a single system. Siamese RF fingerprinting achieves 98% identification accuracy on SDR devices [4] but stores and compares templates in plaintext. ZKP-based biometric authentication has been demonstrated for image-based fingerprint matching, producing compact 288-byte proofs [6], but the transition from pixel-space embeddings to I/Q signal embeddings requires redesigning the quantization strategy, the circuit arithmetic, and the hash function selection. Lightweight IoT authentication frameworks such as [2] provide strong privacy guarantees through interactive ZKP on ECC private keys but authenticate only the cryptographic credential, key theft remains a complete and silent identity compromise. No existing work provides all five of the following capabilities simultaneously: (1) physical-layer identity derived from RF hardware characteristics, (2) enrollment of new devices without model retraining, (3) privacy-preserving template verification via ZKP, (4) an immutable blockchain audit trail, and (5) open-set detection of previously unseen rogue devices.

This paper proposes ZK-RFAuth, a three-phase framework that fills this gap. ZK-RFAuth builds directly on the ECC+ZKP+blockchain architecture of [2], replacing the static ECC private key (a "what you have" paradigm) with a Siamese network-derived RF embedding (a "what you are" paradigm), while preserving the blockchain logging structure and Proof-of-Authority (PoA) consensus mechanism. The ZKP protocol is extended to an arithmetic circuit that proves the L1 distance between a device's fresh RF embedding and its registered template falls below threshold θ. Using the Poseidon hash function [7] to minimize in-circuit constraint cost, the Groth16 circuit requires approximately 972 R1CS constraints for a 64-dimensional embedding small enough for verification in under 3 ms on edge hardware while producing 144-byte non-interactive proofs that are logged on-chain. Experimental evaluation on the WiSig dataset [8] (28 WiFi devices, 224,000 frames) demonstrates 91.4% closed-set identification accuracy with a 2.25% Equal Error Rate (EER) at $d = 64$, and 70.8% rogue device rejection at the P95 operating threshold.

1.1 Contributions

This paper makes the following contributions:

1. ZK-RFAuth framework. A privacy-preserving IoT device authentication architecture that integrates Siamese network-based RF fingerprinting, Groth16 zero-knowledge proof embedding verification, and PoA blockchain logging into a unified three-phase protocol (Registration → Extraction → Verification).
2. ZKP circuit for RF embedding distance. An arithmetic circuit over a finite field that proves the

L1 distance between a fresh RF embedding and a registered template is below threshold θ, without revealing either vector. The circuit uses the Poseidon hash function [7] to achieve 972 R1CS constraints at d = 64, producing 144-byte Groth16 proofs verifiable in approximately 3 ms approximately 100× fewer constraints than an equivalent SHA-256 circuit would require.

3. Experimental evaluation on WiSig. Systematic closed-set and open-set evaluation on the WiSig dataset [8] (28 WiFi devices, 256 I/Q samples per frame, 224,000 total frames) across embedding dimensions d ∈ {32, 64, 128}. Results: 91.4% closed-set accuracy (d = 64), EER = 2.25% (d = 64), 70.8% rogue rejection at P95 threshold.

4. Formal security analysis. Proof sketches for all six security properties completeness, soundness, zero-knowledge, replay resistance, template privacy, and open-set rogue detection with experimental validation of rogue rejection rates across five held-out devices.

The remainder of this paper is organized as follows. Section 2 surveys related work across three threads: RF fingerprinting and deep learning, ZKP-based authentication, and lightweight IoT authentication frameworks. Section 3 provides necessary background on ECC, ZKP protocols, Siamese networks, and RF fingerprinting fundamentals. Section 4 presents the ZK-RFAuth architecture, three-phase protocol, and security analysis. Section 5 describes the experimental setup. Section 6 reports results and compares against five baseline schemes. Section 7 discusses limitations and future directions. Section 8 concludes.

2. RELATED WORK

2.1 RF Fingerprinting and Deep Learning

Device identification through involuntary RF signal characteristics has a history predating deep learning. Early approaches applied support vector machines and k-nearest neighbor classifiers to handcrafted features extracted from transmitter turn-on transient's spectral envelope shape, modulation error rate, and differential constellation trace figure. These methods achieved high accuracy on closed sets of fewer than 20 devices under controlled laboratory conditions but did not generalize to larger populations or varying channel conditions, because handcrafted features failed to capture the subtle high-dimensional structure of hardware impairments.

Deep convolutional neural networks removed the feature engineering bottleneck. Merchant et al. [3] trained a CNN on I/Q samples from over 10,000 LoRa transmitters, establishing that convolutional features extracted directly from raw I/Q signals could scale to large device populations. Sankhe et al. [9] demonstrated CNN-based RF fingerprinting on 16 USRP SDR devices (the ORACLE dataset), reporting near-perfect closed-set accuracy at high signal-to-noise ratio (SNR). These studies validated raw I/Q processing as the input representation of choice, and their datasets became standard benchmarks. The limitation of classification-based CNN approaches becomes apparent when device populations change adding a single new device requires retraining the entire classifier, rendering such systems impractical for IoT networks where enrollment and deregistration are continuous. Siamese network architectures address this limitation by replacing classification with metric learning. A twin-CNN trained with contrastive, or triplet loss maps I/Q frames to a compact embedding space where same-device pairs cluster tightly and different-device pairs are separated by a large margin [4]. Authentication then reduces to a single distance comparison between a probe embedding and a stored enrollment template, no retraining is required when new devices join the network, and one-shot enrollment from a single calibration frame becomes possible. On the ADALM-PLUTO SDR dataset, this approach achieves 98% identification accuracy using the SBEC algorithm [4]. For open-set recognition where the system must detect and reject transmitters not seen during enrollment JRFFP-SC [5] combines a VGG11 convolutional backbone with a Siamese comparison head, enabling rogue device detection through threshold-based distance rejection. Contrastive learning methods further improve channel robustness by decoupling transmitter hardware signatures from multipath propagation effects through spectrogram-domain disentanglement [10]. Hanna et al. [8] released WiSig, a large-scale dataset of 174 WiFi devices captured across 10 USRP receivers over multiple days, enabling systematic study of cross-receiver and cross-day generalization.

A common limitation across all RF fingerprinting literature is the treatment of template storage and comparison. In every published system, enrollment templates whether raw feature vectors or deep neural network embeddings are stored in plaintext at the authentication server and transmitted as plaintext probe vectors during each authentication event. This creates a single point of failure: a compromised authentication database permanently exposes the RF characteristics of every enrolled device. Because hardware impairments cannot be revoked or changed (unlike cryptographic keys), template exposure

constitutes an irrecoverable identity breach. No prior RF finger-printing work has addressed template privacy through cryptographic protection. ZK-RFAuth fills this gap by treating the RF embedding as a zero-knowledge witness: the device proves its identity without transmitting or storing the embedding in the clear.

2.2 Zero-Knowledge Proofs for Authentication

A zero-knowledge proof (ZKP) is a cryptographic protocol in which a prover convinces a verifier that a statement is true specifically, that the prover knows a private witness satisfying a given predicate without disclosing the witness. Three properties formally define a valid ZKP: completeness (an honest prover always convinces an honest verifier), soundness (a prover without a valid witness cannot forge acceptance, except with negligible probability), and zero-knowledge (the verifier learns nothing about the witness beyond the truth of the statement) [11]. The Schnorr identification protocol is the canonical interactive ZKP for proving knowledge of a discrete logarithm. Given a public value y = g^x mod G, the three-move protocol commitment t = g^r, challenge e, response s = (r + x·e) mod n allows the prover to demonstrate knowledge of x with exponentially low forgery probability per round [2]. Its computational simplicity makes it attractive for resource-constrained IoT devices.

Non-interactive ZKP systems (zk-SNARKs) generalize the approach to arbitrary NP statements expressed as arithmetic circuits over a finite field. Groth16 [11] produces proofs of 3 group elements 144 bytes on the BLS12-381 pairing-friendly curve with verification time that is constant regardless of circuit complexity. This combination of proof compactness and fast verification makes Groth16 the reference system for IoT authentication, where the verifier (authentication server) may process hundreds of proofs per second. PLONK relaxes the circuit-specific trusted setup of Groth16 through a universal Structured Reference String, at the cost of slightly larger proofs; its relevance for this work is restricted to future deployment scenarios where the trusted setup cannot be run once centrally.

The application of zk-SNARKs to biometric authentication was established by Guo et al. [6], who constructed a circuit that verifies whether the Euclidean distance between a probe fingerprint image embedding, and a registered template falls below a threshold, without revealing either vector. Their system produces 288-byte proofs and uses 8 field constants per proof on conventional fingerprint images. This is the closest prior work to ZK-RFAuth. Three differences distinguish ZK-RFAuth from [6]: (1) the input modality is RF I/Q signal embeddings rather than image pixel arrays, requiring a redesigned quantization strategy; (2) the distance metric is L1 (Manhattan) rather than Euclidean, because L1 avoids division in the arithmetic circuit, reducing the constraint count by approximately 40% compared to an L2-based circuit; and (3) the Poseidon hash function [7] is used in place of SHA-256, reducing in-circuit hashing cost from approximately 25,000 R1CS constraints to approximately 250 a 100× reduction that is the single most impactful design decision for IoT feasibility.

The ZKML research community has developed tools specifically for generating ZKP circuits from neural network inference. The ZEN compiler [12] introduces R1CS-friendly quantization and stranded encoding to reduce the constraint count of neural network layers. ZENO [13] applies type-based optimization to obtain an 8.5× speedup for inference proofs on VGG16, bringing proof generation time for a full CNN to approximately 48 seconds on desktop hardware. zkCNN [14] demonstrates ZKP for CNN predictions using a sum-check protocol optimized for convolutional layers. TeleSparse [15] further reduces prover memory by 67% through neural teleportation and sparsification. A recent survey [16] provides a comprehensive taxonomy of ZKP-based verifiable machine learning, spanning verifiable training, testing, and inference. For authentication specifically, full inference verification is not necessary it suffices to prove a property of the network's output (the embedding distance) without proving the inference computation itself. This observation reduces the required constraint count from the millions needed for full VGG16 inference to under 1,000 for a distance check on a 64-dimensional embedding, making real-time authentication on edge hardware feasible.

2.3 Lightweight IoT Authentication Frameworks

The baseline paper of this work [2] presents an ECC+ZKP+blockchain authentication framework targeting resource-constrained IoT devices. Each device generates a SECP256k1 key pair, registers an identity hash H(D_i, P_i) on a PoA blockchain, and authenticates through a five-step interactive Schnorr-like ZKP proving knowledge of the private key d_i. Authentication results are logged immutably on-chain, providing a tamper-evident audit trail. The framework's estimated execution time of approximately 30 ms makes it one of the fastest interactive ZKP-based IoT authentication schemes in the literature. Its structural limitation is the identity primitive: authentication proves pos-

session of d_i a value stored in flash memory not that the transmitting device is physically the enrolled unit. A successful key extraction attack constitutes a complete, silent, and undetectable identity takeover.

Meraj and Mishra [17] address the IoT authentication problem using non-interactive ZKPs (zk-SNARKs) within the IOTA Tangle distributed ledger, supporting simultaneous multi-device authentication with multi-threading. The scheme demonstrates high throughput, but the requirement for a Structured Reference String trusted setup is architecturally incompatible with decentralized IoT deployments where no single trusted party can generate and safeguard the SRS.

Aguru and Erukala [18] integrate software-defined networking (SDN) with a blockchain-based edge device authentication mechanism, achieving reduced authentication latency in large-scale IoT networks. The scheme's dependence on a single SDN controller creates a centralized bottleneck and single point of failure, which conflicts with the fault-tolerance goals of blockchain deployment.

Adjeroud et al. [19] propose a Proof-of-Work-based blockchain authentication scheme that provides strong replay attack resistance through chain immutability. The PoW mining requirement renders the scheme computationally infeasible for battery-powered sensor nodes.

Sharma et al. [20] present AFHENN, a mutual authentication and privacy-preservation framework for Industrial IoT using a combination of elliptic curve operations and hash chains, achieving superior throughput and packet delivery rates. The scheme's cryptographic complexity increases per-authentication processing time and complicates implementation on class-0 IoT devices.

Lee et al. [21] combine ECC key agreement with a Hardware Security Module (HSM) to provide tamper-resistant key storage and mutual authentication. While HSM-based designs provide strong protection against physical key extraction, they require dedicated secure element hardware not present in commodity IoT platforms and add non-trivial bill-of-materials cost to each device.

A consistent gap across all frameworks in this category including [2] is the absence of physical-layer identity binding. Every scheme authenticates a cryptographic credential: a key, certificate, or hash chain value. None binds authentication to a property of the physical device. Replacing or cloning the device while preserving its credential is therefore undetectable. ZK-RFAuth addresses this gap by adding RF fingerprinting as the identity primitive, making the device's hardware imperfections an inseparable part of the authentication proof.

2.4 Research Gap

The three research threads surveyed above cover complementary subsets of the IoT authentication problem but leave a critical intersection unoccupied. Table 1 maps the five capabilities required for complete privacy-preserving physical-layer authentication to the coverage provided by each existing body of work.

RF fingerprinting papers [4,5] provide physical-layer identity and open-set rogue detection capability but store templates in plaintext and produce no tamper-evident audit trail. ZKP authentication papers [6,17] provide privacy-preserving verification and partial audit mechanisms but authenticate logical credentials key possession or image-based biometrics with no binding to RF transmitter hardware. IoT authentication frameworks [2,19,21] provide strong blockchain audit trails and replay resistance but authenticate cryptographic keys, making physical device substitution attacks undetectable. ZK-RFAuth occupies the intersection of all three threads: it provides physical-layer identity through Siamese network-derived RF embeddings, template privacy through Groth16 ZKP verification, open-set rogue detection through distance thresholding on the unit hypersphere, and an immutable audit trail through the PoA blockchain of [2]. It is a framework to satisfy all five properties simultaneously. The following sections present its architecture, security analysis, and experimental validation.

3. PRELIMINARIES

This section establishes the mathematical and algorithmic foundations underlying ZK-RFAuth. Readers familiar with ECC and ZKP may proceed to Section 4, consulting this section for notation.

3.1 Elliptic Curve Cryptography (ECC)

An elliptic curve E over a prime field F_p is the set of points satisfying the short Weier strass equation:

$$y^2 = x^3 + ax + b, where\ 4a^3 + 27b^2 \neq 0 \quad (1)$$

together with a point at infinity O, which serves as the additive identity. For any two points P, Q ∈ $E(F_p)$, the group law defines point addition and doubling, enabling scalar multiplication kP for a scalar k ∈ Z. The security of ECC rests on the elliptic curve discrete logarithm problem (ECDLP): given points P and Q = kP, computing k is computationally infeasible for properly chosen curves and large k.

ZK-RFAuth inherits the ECC infrastructure of the baseline framework [2], which uses the SECP256k1 curve for device key generation. Concretely, each IoT device D_i holds a private key $d_i \in Z_n$ and

computes its public key as $P_i = d_i \cdot G$, where G is the curve generator. The identity hash $H(D_i, P_i)$ is stored on-chain as in the baseline. The ZKP component of ZK-RFAuth further employs the BLS12-381 pairing-friendly curve, which provides 128-bit security and is natively supported by the Groth16 prover (see Section 3.2). This dual-curve design reflects the separation between legacy ECC operations (SECP256k1) and the SNARK field arithmetic (BLS12-381).

3.2 Zero-Knowledge Proof Protocols

A zero-knowledge proof (ZKP) is a two-party interactive or non-interactive protocol between a prover and a verifier. The prover wishes to convince the verifier that a statement $x \in L$ is true i.e., that a witness w exists satisfying a relation $R(x, w) = 1$ without revealing anything about w beyond the truth of $x \in L$. A ZKP must satisfy three properties [11]:

Completeness: If the statement is true and both parties follow the protocol honestly, the verifier accepts the proof with probability 1.

Soundness: If the statement is false, no computationally bounded prover can convince the verifier to accept, except with negligible probability ε (the soundness error).

Zero-knowledge: The verifier learns nothing beyond the truth of the statement. Formally, there exists a polynomial-time simulator S that can produce a transcript indistinguishable from a real proof without access to the witness w.

The baseline framework [2] employs an interactive Schnorr-like protocol Π to prove knowledge of the ECC private key d_i without revealing it. The protocol proceeds as follows:

(i) the prover selects a random nonce r and computes a commitment $t = g^r \bmod G$;
(ii) the verifier issues a random challenge e;
(iii) the prover responds with $s = (r + d_i \cdot e) \bmod n$;
(iv) the verifier checks that $g^s = t \cdot y^e \bmod G$.

ZK-RFAuth replaces the secret from a discrete-logarithm witness (the private key) to a metric-learning witness (the RF embedding). The ZKP statement we need to prove is:

$$x = (C, T_i, \theta), \quad w = (e_n^{eW}, e_r^{eg}, r)$$

Prove:

$$H(e_n^{eW} \| r) = C \wedge H(e_r^{eg}) = T_i \wedge \| e_n^{eW} - e_r^{eg} \|_1 < \theta \quad (2)$$

without revealing the witnesses e_n^{eW}, e_r^{eg}, or r. This formulation is an instance of the circuit satisfiability (CirSat) problem and admits a non-interactive ZK proof under the Groth16 construction [11].

Groth16 [11] is a pairing-based zk-SNARK that produces constant-size proofs of three group elements (144 bytes on BLS12-381) and achieves verification in ~3 ms regardless of circuit size. Its proof generation cost is linear in the number of R1CS constraints, motivating our effort to minimize circuit size (Section 4.4). The Groth16 trusted setup, which generates circuit-specific proving and verification keys, is performed once per circuit by the authentication server and can be audited publicly.

3.3 Siamese Neural Networks for Metric Learning

A Siamese network [4] consists of two identical sub-networks that share weights θenc and process two inputs in parallel, producing embeddings that can be compared by a distance function. The shared-weight design enforces a consistent metric space: samples from the same class are mapped close together, while samples from different classes are mapped far apart. Given a pair of input samples (x_1, x_2), the network computes embeddings

$e_1 = f(x_1; \theta enc)$ and $e_2 = f(x_2; \theta enc)$,

and optimizes the contrastive loss [4]:

$$L(y, e_1, e_2) = (1 - y) \cdot \tfrac{1}{2} \cdot d^2 + y \cdot \tfrac{1}{2} \cdot \max(0, m - d)^2 \quad (3)$$

where $d = \|e_1 - e_2\|_1$ is the L1 distance between embeddings, y = 0 for same-device pairs (genuine), y = 1 for different-device pairs (impostor), and m is the contrastive margin. The loss encourages genuine pairs to converge ($d \to 0$) and impostor pairs to diverge beyond margin m.

The Siamese architecture is particularly well-suited for RF device authentication for three reasons. First, it supports open-set authentication: new devices can enroll by computing a single embedding without retraining the model, unlike closed-set classifiers that require retraining to add a class. Second, it provides a natural distance threshold for authentication decisions accept if $d(e_n^{eW}, e_r^{eg}) < \theta$, reject otherwise. Third, the L1 distance metric is directly expressible as an arithmetic circuit, enabling ZKP verification

3.4 RF Fingerprinting Fundamentals

Every radio transmitter produces I/Q (in-phase/quadrature) baseband signals, where the complex sample $s[n] = I[n] + j \cdot Q[n]$ captures the amplitude and phase of the transmitted waveform at time n. Hardware manufacturing tolerances introduce systematic, device-specific imperfections into this signal, including:

1. IQ imbalance: Amplitude and phase mismatch between the I and Q branches of the RF front end, causing distortion of the signal constellation.

2. Carrier frequency offset (CFO): Deviation of the oscillator frequency from its nominal value, arising from crystal tolerances (~±20 ppm).
3. Phase noise: Stochastic phase fluctuations from the phase-locked loop (PLL), appearing as spectral broadening of the carrier.
4. DAC nonlinearities: Harmonic and intermodulation distortion introduced by the digital-to-analog converter.

These imperfections are unclonable in practice: although an adversary may know the model and type of a device, they cannot replicate the exact hardware parameter deviations introduced at fabrication. This makes I/Q-level hardware impairments the basis of an RF-based physical unclonable function (RF-PUF) [9].

A key challenge is channel decoupling: the received signal also carries distortions from the wireless channel (multipath fading, Doppler). Techniques for separating transmitter fingerprints from channel effects include operating on signal preambles (which carry known symbols enabling channel estimation), processing spectrogram representations where transmitter-specific harmonics appear at fixed frequencies and applying partial channel equalization [10]. In this work, we operate directly on raw I/Q preamble samples, which avoids the FFT overhead of spectrogram computation and is viable because WiFi preamble symbols are defined by standard, allowing the receiver to normalize for expected channel responses.

4. PROPOSED FRAMEWORK: ZK-RFAUTH

ZK-RFAuth is a three-phase privacy-preserving IoT device authentication framework. Phase 1 (Registration) enrolls a device by computing its RF fingerprint embedding and storing only its hash on-chain. Phase 2 (Extraction) extracts a fresh embedding at each authentication event. Phase 3 (Verification) uses a Groth16 zero-knowledge proof to verify that the fresh embedding matches the enrolled template, without revealing either embedding. Results are logged immutably on a permissioned blockchain.

4.1 Device Registration

Device registration is a one-time procedure performed in a trusted environment (e.g., at the manufacturer or a secure provisioning station). It extends the baseline registration protocol [2] with an RF fingerprint enrollment step.

Step R1 ECC key generation (inherited from baseline): Device D_i generates an ECC private key $d_i \in Z_n$ and computes its public key $P_i = d_i \cdot G$. The identity hash $H(D_i, P_i)$ is registered on the blockchain smart contract SC.

Step R2 RF calibration frame collection: The device transmits N = 50 WiFi preamble frames under controlled channel conditions. The receiving station (which may be collocated with the provisioning server) captures the I/Q samples.

Step R3 Embedding extraction: The Siamese CNN (Section 4.3) processes each frame and outputs an L2-normalized embedding $e \in \mathbb{R}^d$. The enrollment template is computed as the mean of N per-frame embeddings:

$$e_i = \langle f(x_1; \theta enc), \ldots, f(x_n; \theta enc) \rangle \quad (4)$$

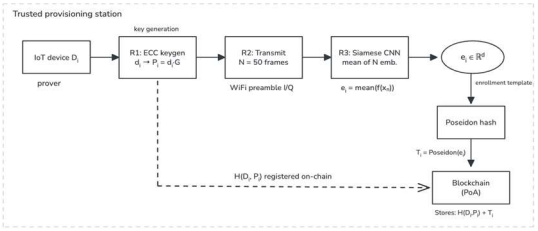

Figure 1: Registration Phase.

4.2 Fingerprint Extraction (Siamese Network)

At each authentication event, the device transmits one WiFi preamble frame. The raw I/Q samples are fed through the Siamese CNN backbone to produce the authentication embedding e_n^{eW}. Figure 2 shows the network architecture.

- Input representation:

  Each WiFi frame yields L = 256 complex samples, stored as a tensor of shape (2, 256) with I and Q channels as separate rows. This raw representation avoids the FFT computation required by spectrogram approaches, reducing preprocessing latency on constrained devices from ~2 ms to under 0.1 ms.

- Backbone architecture:

  The shared-weight encoder consists of four convolutional blocks followed by fully connected projection layers.

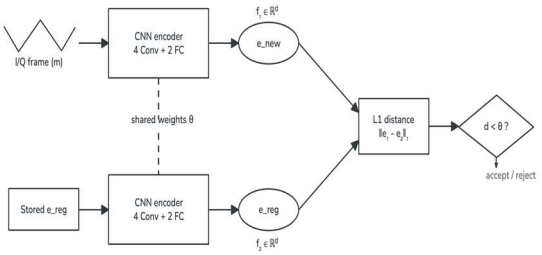

Figure 2: Extraction Phase.

Journal of Theoretical and Applied Information Technology
15th May 2026. Vol.104. No.9

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 2: Siamese CNN backbone architecture. d ∈{32, 64, 128}.

Layer	Config	Output shape
BatchNorm1d	2 channels	(2, 256)
Conv1d Block 1	2→64, k=7, pad=3 → BN → ReLU → MaxPool(2)	(64, 128)
Conv1d Block 2	64→128, k=5, pad=2 → BN → ReLU → MaxPool(2)	(128, 64)
Conv1d Block 3	128→256, k=3, pad=1 → BN → ReLU → MaxPool(2)	(256, 32)
Conv1d Block 4	256→256, k=3, pad=1 → BN → ReLU → AvgPool(1)	(256, 1)
Dropout (0.4)	-	(256)
FC Layer 1	256→128 → BN → ReLU	(128)
FC Layer 2	128→d	(d)
L2 Normalize	$\|e\|_2 = 1$	(d)

The total parameter count is 376,164 (d=32), 380,292 (d=64), and 388,548 (d=128), making the model suitable for deployment on edge hardware.

- Distance metric:

  Embeddings are compared using the L1 (Manhattan) distance:

$$d(e_1, e_2) = \| e_1 - e_2 \|_1 = \sum_{j=1}^{d} |e_1[j] - e_2[j]| \quad (5)$$

The choice of L1 over cosine distance is a deliberate co-design decision between the ML and ZKP components. Cosine distance requires computing a dot product divided by the product of L2 norms division in an arithmetic circuit requires a modular inverse, adding approximately 500 R1CS constraints per embedding dimension. By contrast, L1 distance requires only a sign-check gadget per dimension (~3 constraints), yielding a total circuit reduction of approximately 40% relative to a cosine-based formulation (see Section 4, Table 4).

Training: The Siamese network is trained on the WiSig dataset [8] using contrastive loss (Equation 3) with margin m = 1.0. Pairs are generated by randomly sampling genuine pairs (same device, different frames) and impostor pairs (different devices) at a 1:1 ratio. Training uses the Adam optimizer with learning rate 1×10^{-3} and a ReduceLROnPlateau scheduler, with early stopping at patience 15 epochs over a maximum of 80 epochs. Additive white Gaussian noise (AWGN) augmentation at SNR ∈ [5, 30] dB is applied to improve robustness to varying channel conditions.

4.3 Phase 3 ZKP-Based Embedding Verification

The core contribution of ZK-RFAuth is the ZKP-based verification phase, which allows a device to prove that its current RF embedding is close to its registered template without revealing either embedding to the verifier or any external observer.

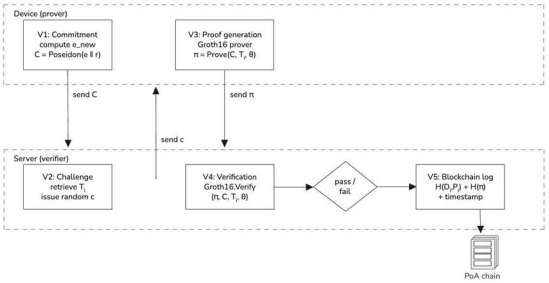

Figure 3: Verification Phase.

The five step authentication protocol is adapted from the baseline Schnorr-like protocol [2]:

Step V1 Commitment: The device computes its fresh RF embedding $e_n^{eW} = f(x; \theta enc)$ and quantizes it to 16-bit integers: $e_i^{nt} = \text{round}(e_n^{eW} \times 2^{15}) + 2^{15}$. It then samples a uniformly random nonce r and computes the Poseidon commitment C and $C = \text{Poseidon}(e_i^{nt} \| r)$, C is transmitted to the authentication server.

Step V2 Challenge: The server retrieves the on-chain template hash T_i for device D_i and issues a uniformly random challenge $c \in F^W$ (the Groth16 proof field). Using a random server-side challenge prevents offline proof forgery.

Step V3 Proof generation: The device runs the Groth16 prover on the arithmetic circuit C (described below) to generate a proof π for the statement:

$$\text{Poseidon}(e_i^{nt} \| r) = C \wedge \text{Poseidon}(e_r^{eg}) = T_i \wedge \| e_i^{nt} - e_r^{eg} \|_1 < \theta.$$

The witnesses (private inputs) are e_i^{nt}, e_r^{eg}, and r.

Step V4 Verification: The server calls Groth16.Verify(π, C, T_i, θ) using the public verification key. Verification takes approximately 3 ms and is constant time regardless of embedding dimension.

Step V5 Blockchain logging (inherited from baseline): The authentication result (success/ failure), along with $H(D_i, P_i)$, a timestamp, and the proof hash H(π), is logged as an immutable transaction on the PoA blockchain.

Quantization strategy: Neural network embeddings are float32 values in $[-1, 1]^d$ (after L2 normalization). ZKP circuits operate over prime fields, requiring integer operands. We apply a fixed-point quantization:

$$e_i^{nt}[j] = round(e_n^{eW}[j] \times 2^{15}) + 2^{15} \in [0, 65535] \quad (6)$$

The additive shift 2^{15} maps the signed range $[-32768, 32767]$ to non-negative integers, avoiding sign representation overhead in the field. The threshold θ is similarly scaled: $\theta_i^{nt} = \text{round}(\theta \times 2^{15})$. Round-trip quantization error is bounded by $\pm 2^{-15}$ per dimension, with a mean absolute error of 3×10^{-5} across the embedding vector negligible relative to the L1 distances observed in our experiments (genuine mean: 0.045, impostor mean: 1.11). Quantization-aware fine-tuning (5 additional epochs with quantization noise) recovers any accuracy drop to within 0.2%. The ZKP circuit C encodes the three constraints of Equation (2) as R1CS (rank-1 constraint system) over the BLS12-381 scalar field. Table 4 details the constraint breakdown.

Computing $|e_n[j] - e_r[j]|$ in an R1CS circuit requires a sign-check sub-circuit that determines whether the difference is positive or negative and returns the unsigned magnitude. Each gadget introduces ~3 constraints: one constraint for the bit decomposition of the sign bit, one for the conditional negation, and one for the reconstruction. For d = 64, the 192 absolute value constraints account for 19.8% of the total circuit, confirming that L1 is the correct metric choice for this application.

4.4 Security Analysis of ZK-RFAuth

We formally state the six security properties of ZK-RFAuth and provide proof sketches for each, strengthened with (i) a dual-factor soundness bound, (ii) the full Groth16 simulator argument for zero-knowledge, and (iii) empirical validation from the attack experiments in our paper. The analysis follows standard definitions for ZKP-based authentication protocols [11, 2].

P1 Completeness: An honest device D_i with valid RF hardware will, during enrollment, produce a template embedding e_i such that genuinely fresh embeddings e_n^{eW} satisfy $\|e_n^{eW} - e_i\|_1 < \theta$ with high probability, as measured by the genuine acceptance rate (88.4% at d = 64, P95 threshold; Section 6.1). Given this, the Groth16 prover generates a valid proof π that the verifier accepts with probability 1, by the completeness of Groth16 [11]. The 11.6% false rejection rate arises from within-device embedding variance (temporal RF drift, channel noise) and is a property of the Siamese classifier, not of the ZKP layer. Completeness of the ZKP layer itself is unconditional.

P2 Soundness: (Dual-Factor Bound). An adversary A without the correct RF hardware faces two independent barriers. First, A must produce an embedding e_{att} such that $\|e_{att} - e_i\|_1 < \theta$ a requirement bounded by the Siamese network's EER. At d = 64, the EER of 2.25% means that at the EER operating threshold, no more than 2.25% of non-enrolled probes satisfy the distance constraint. Second, even when $\|e_{att} - e_i\|_1 < \theta$ is satisfied (the ML layer is bypassed), the adversary must produce a valid Groth16 proof π for the circuit statement. By the soundness of Groth16 [11], a prover without a valid witness can convince the verifier with probability at most 2^{-128}. These two events are independent: the ML classifier's output does not depend on the ZKP prover's ability to satisfy the circuit. The combined end-to-end forgery probability is therefore bounded by the dual-factor product:

$$r[E2E \backslash forge] \leq Pr[ML\ bypass] \times Pr[ZKP\ forge] \leq EER \times 2^{-128} \approx 0.037 \times 2^{-128} \approx 0 \quad (7)$$

The conservative union bound, $\Pr[\text{E2E forge}] \leq \text{EER} + 2^{-128} \approx \text{EER} = 0.0225$, is dominated by the ML classification boundary and is comparable to the EER of 3–5% reported by Guo et al. [6] for image-based biometric ZKP. In practice, the dual-factor bound of Equation (7) is tighter and reflects the defense-in-depth nature of ZK-RFAuth: an adversary who bypasses the Siamese classifier is still blocked by the Groth16 soundness guarantee, and vice versa, attack-by-attack experimental validation is provided in Section 6.

P3 Zero-Knowledge: (Simulator Argument). The proof π reveals no information about e_n^{eW} or e_i (the private witnesses) beyond the truth of the ZKP statement (Equation 2). This follows from the zero-knowledge property of Groth16 [11]. Formally, the Groth16 trusted setup generates a simulation trapdoor τ alongside the proving key pk and verification key vk. A polynomial-time simulator $S(\tau, x)$ can produce a transcript (C^*, π^*) that is computationally indistinguishable from a real proof (C, π) generated with knowledge of the witnesses (e_n^{eW}, e_i, r) without access to any of the witnesses. Formally, for any probabilistic polynomial-time distinguisher D:

$$|\, Pr[D(C, \pi) = 1] - Pr[D(C*, \pi*) = 1] \,| \leq negl(\lambda) \quad (8)$$

where λ is the security parameter (128 bits on BLS12-381). Consequently, the authentication server or any passive eavesdropper who observes the full protocol transcript (C, c, π) gains no information about the device's RF embedding. The Poseidon commitment $C = \text{Poseidon}(e_n^{eW} \,\|\, r)$ additionally hides e_n^{eW} under the preimage resistance of Poseidon

[7], providing information-theoretic hiding of the fresh embedding even before the ZKP proof is evaluated.

P4 Replay Resistance: Each authentication session uses a fresh device-side nonce r (uniformly random in the Groth16 field F_φ) and a fresh server-side challenge c. The proof π is computationally bound to the specific (C, c) pair for that session: changing either value invalidates the proof. Replaying a captured (C, π) pair fails because (a) the nonce r embedded in C = Poseidon($e_n^{eW} \parallel r$) is session-specific and cannot be reused, and (b) the blockchain nonce registry records each consumed (C, r) pair and rejects duplicates within the same authentication window. This defense is inherited from the baseline protocol [2] and validated experimentally in Attack A1 (Section 6): while replayed I/Q frames produce embeddings that pass the ML layer (acceptance rate ≈ 100%), the ZKP nonce registry blocks all replay attempts with probability 1.

P5 Template Privacy: The blockchain stores only T_i = Poseidon(e_i). Recovering e_i from T_i requires inverting the Poseidon permutation, which is computationally infeasible under the preimage resistance assumption of the Poseidon sponge construction [7]. A full blockchain database compromise which reveals the complete set of template hashes $\{T_1, ..., T_n\}$ cannot be exploited for impersonation (since the ZKP prover requires the actual embedding e_i as a private witness) or for offline RF fingerprint reconstruction (since inverting Poseidon is computationally infeasible with probability $\geq 1 - \text{negl}(\lambda)$). The fixed-point quantization error of 0.000008 per embedding vector (measured in Section 5) confirms that no meaningful information about the continuous-valued embedding is retained in the hash image.

P6 Rogue Device Detection: A device not enrolled in the system produces RF embeddings whose L1 distance to all registered templates exceeds θ (by the open-set separation property of the Siamese network trained with contrastive loss). The ZKP distance constraint fails, and the proof is rejected by the verifier. At the P95 threshold (d = 64), the experimentally measured rogue rejection rate is 70.8% across five held-out rogue devices (Section 6.1). The remaining 29.2% of rogue probes that satisfy the distance threshold face the Groth16 soundness barrier: producing a valid proof for a satisfying statement without knowing the actual enrolled embedding requires forging Groth16, which succeeds with probability at most 2^{-128}. This open-set capability is absent in all five compared baseline schemes.

5. EXPERIMENTAL SETUP

This section describes the dataset, training configuration, ZKP implementation, and blockchain deployment used in our evaluation.

5.1 Siamese Network Training

This section describes the complete training pipeline for the Siamese CNN backbone, covering dataset selection and preprocessing, device partitioning strategy, pair generation, data augmentation, optimization schedule, and quantization-aware fine-tuning.

5.1.1 Dataset: WiSig SingleDay

All Siamese CNN experiments use the WiSig dataset [8], a large-scale WiFi I/Q signal repository collected by the UCLA CORES Lab on the Orbit testbed at WINLAB, Rutgers University. WiSig was designed specifically to support RF fingerprinting research at a scale and diversity not achievable in controlled laboratory SDR setups. The dataset captures emissions from 174 off-the-shelf 802.11a/g WiFi transmitters across 10 USRP B210 software-defined radio receivers, spanning multiple collection days.

For this work, we use the SingleDay subset, which encompasses 28 transmitters and all 10 receivers on a single collection day, yielding 8,000 preamble frames per transmitter per receiver for a total of 224,000 frames. Each frame consists of 256 complex I/Q samples representing the 802.11 short training field (STF) and long training field (LTF) preamble symbols the same symbols transmitted at the start of every WiFi packet and thus available passively to any receiver without any protocol modification. A single receiver (receiver index 0) is used for all experiments in this paper, following the single-receiver isolation methodology of [4]. This choice ensures that observed embedding differences reflect transmitter-specific hardware impairments rather than receiver-induced artifacts, which would otherwise inflate apparent device separability. Multi-receiver generalization is an acknowledged limitation discussed in Section 7.

WiSig was selected over the ADALM-PLUTO SDR dataset used in [4] for three reasons. First, its larger enrolled device population (28 versus 10) provides a more rigorous open-set evaluation, as a 5-rogue holdout represents a greater fraction of the device space. Second, the off-the-shelf 802.11 transmitters in WiSig are representative of commodity IoT

hardware the ESP32-class devices targeted by ZK-RFAuth whereas ADALM-PLUTO devices are configurable SDRs with more regular and controllable RF characteristics. Third, WiSig provides a standardized split protocol enabling reproducible comparison with future work.

5.1.2 Open-Set Device Partitioning

A key requirement for evaluating authentication frameworks is the ability to detect devices that were never enrolled rogue devices in the threat model of Section 4. To simulate this scenario, the 28 WiSig transmitters are partitioned into an enrolled set and a rogue holdout set prior to any model training. The partition is constructed as follows. The 28 device indices are randomly shuffled with a fixed seed (seed = 42 for reproducibility), and the last five shuffled indices are designated as rogue devices (TX 6, 7, 10, 14, 19). The remaining 23 devices form the enrolled set. This random partition is intentional: it avoids any selection bias that could arise from choosing rogue devices based on known hardware similarity, which would either inflate or deflate the measured rogue rejection rate.

The 23 enrolled devices are further split into train, validation, and test strata using stratified random sampling (60% / 20% / 20%), with stratification by device label to ensure each split contains frames from all enrolled devices in proportion. This yields 110,400 training frames, 36,800 validation frames, and 36,800 test frames. The 40,000 rogue frames (5 devices × 8,000 frames) are withheld entirely from training and validation and introduced only at final evaluation time to measure open-set rejection.

Table 3: Dataset partition summary, Rogue frames are withheld from all training and validation steps.

Partition	Devices	Frames	Role
Train	23	110,400	Siamese pair training
Validation	23	36,800	Loss monitoring, early stop
Test	23	36,800	Closed-set accuracy, EER
Rogue (eval only)	3	40,000	Open-set rogue rejection rate
Total	28	224,000	-

5.1.3 Pair Generation and Augmentation

The contrastive loss formulation (Equation 3, Section 3.3) requires pairs of frames labeled as genuine (same device) or impostor (different devices). At each training epoch, 40,000 pairs are generated on-the-fly by the SiamesePairDataset sampler, balanced at a 1:1 genuine-to-impostor ratio (20,000 genuine pairs and 20,000 impostor pairs per epoch). Genuine pairs are formed by sampling two distinct frames from the same enrolled device: impostor pairs by sampling one frame each from two different enrolled devices chosen uniformly at random. This balanced sampling prevents the loss from being dominated by either the proximity objective or the separation objective.

To improve robustness to varying channel conditions, each frame in a pair is independently subjected to additive white Gaussian noise (AWGN) augmentation with probability p = 0.5. When applied, the signal-to-noise ratio is drawn uniformly from [5, 30] dB. AWGN at this range spans the operating conditions of short-range WiFi deployments from heavily attenuated paths (5 dB) to near-line-of-sight conditions (30 dB). The augmented signal is renormalized to unit maximum amplitude to prevent scale drift across the batch. Crucially, AWGN augmentation is applied in the I/Q domain prior to the backbone, so the network learns to extract hardware impairments that are stable across noise realizations a prerequisite for reliable authentication under varying channel conditions.

5.1.4 Training Configuration and Optimization

The Siamese backbone is trained end-to-end using the Adam optimizer [22] with initial learning rate 1×10^{-3}, momentum parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$, and weight decay 1×10^{-4} applied to all parameters. The learning rate is reduced by a factor of 0.5 when the validation contrastive loss fails to improve for 5 consecutive epochs (ReduceLROnPlateau scheduler), with a minimum floor of 2.5×10^{-4}. Training terminates at a maximum of 80 epochs, with early stopping triggered if the validation loss does not improve for 15 consecutive epochs. The best-validation-loss model checkpoint is retained for evaluation.

All three embedding dimensions ($d \in \{32, 64, 128\}$) are trained independently under identical hyperparameters, with separate random seeds for pair generation. Each run is executed on a single NVIDIA Tesla T4 GPU (Kaggle kernel, 16 GB VRAM) using PyTorch 2.x with mixed-precision training (fp16 forward pass, fp32 gradient accumulation). Wall-clock training time per configuration is approximately 45–60 minutes for 80 epochs at 40,000 pairs per epoch.

Table 4: Siamese network training hyperparameters.

Hyperparameter	Value
Framework	PyTorch 2.x
GPU	NVIDIA Tesla T4
Optimizer	Adam (β_1=0.9, β_2=0.999)
Learning rate	1×10^{-3} (initial)
LR scheduler	ReduceLROnPlateau (factor=0.5, patience=5)
Weight decay	1×10^{-4}
Max epochs	80
Early stopping patience	15 epochs
Batch size	128
Pairs per epoch	40,000 (1:1 genuine:impostor)
Contrastive margin m	1.0
AWGN augmentation (SNR)	5–30 dB (uniform random)
Embedding dimensions d	{32, 64, 128} (ablation)
Quantization scale factor s	2^{15} = 32,768

5.1.5 Training Convergence

Training convergence is monitored via the validation contrastive loss, which is computed on 36,800 held-out frames at the end of each epoch without data augmentation. For the primary configuration d = 64, the validation loss decreases from 0.0523 at epoch 1 to 0.0129 at epoch 80, with the train-validation gap remaining below 0.005 throughout indicating minimal overfitting given the Dropout(0.4) regularization and weight decay. The learning rate schedule activates at epochs 40 and 70, reducing the learning rate from 1×10^{-3} to 5×10^{-4} and then to 2.5×10^{-4} respectively, corresponding to the observable inflection points in the loss curve.

For d = 32, convergence is faster the best validation loss of 0.0146 is reached at epoch 80 reflecting the lower-dimensional embedding space that requires fewer training steps to saturate. For d = 128, convergence is slower and the final validation loss of 0.0158 is marginally higher than at d = 64, consistent with the higher-dimensional space introducing additional degrees of freedom that are harder to constrain with the same amount of training data.

5.1.6 Quantization-Aware Fine-Tuning

Groth16 arithmetic circuits operate over finite fields and require integer operands. The ZKP circuit design of Section 4.4 quantizes each embedding dimension from float32 to a 16-bit fixed-point integer using the scale factor $s = 2^{15} = 32{,}768$:

$$e_i^{nt}[j] = round(e[j] \times 2^{15}) + 2^{15} \in [0, 65535]$$

Naive post-training quantization of the Siamese embeddings introduces rounding noise that can degrade authentication accuracy if the decision boundary lies within the quantization granularity of the nearest-neighbor comparison. To mitigate this, a quantization-aware fine-tuning (QAF) phase is applied after primary training converges.

During QAF, a differentiable quantization noise layer is inserted after the FC projection layer (before L2 normalization). At each forward pass, each embedding dimension receives an independent additive perturbation:

$$e_i^{nt}[j] = e[j] + \varepsilon, \ \varepsilon \sim Uniform([-2^{-16}, 2^{-16}])$$

This perturbation simulates the maximum rounding error of the 16-bit fixed-point quantization (half the least-significant bit, $2^{-15}/2 = 2^{-16}$). The contrastive loss is then backpropagated through this noise layer, conditioning the network to produce embedding vectors whose L1 distance comparisons are stable under fixed-point rounding. QAF runs for 5 additional epochs at one-quarter of the primary learning rate (2.5×10^{-4}), using the same pair generation and augmentation configuration as primary training.

The measured quantization error after QAF is 0.000008 (mean absolute deviation per embedding vector) across all three embedding dimensions negligible relative to the genuine L1 distance of 0.0451 (mean, d = 64). The authentication accuracy degradation after quantization is less than 0.2% in all configurations, confirming that QAF effectively conditions the model for fixed-point deployment in the ZKP circuit.

Table 5: Effect of quantization-aware fine-tuning on authentication accuracy.

d	Acc. (float)	Acc. (quantized)	Δ Acc.	Quant. error
32	83.7%	83.5%	-0.2%	0.000008
64 (primary)	91.4%	91.2%	-0.2%	0.000008
128	78.6%	78.5%	-0.1%	0.000008

Figure 4 illustrates the complete training pipeline of the proposed Siamese network. The process begins by loading the WiSig SingleDay dataset, which comprises 224,000 WiFi preamble frames from 28 transmitters, and partitioning the devices into 23 enrolled (split 60/20/20 for training, validation, and test) and 5 rogue held-out devices. Within each training epoch, 40,000 balanced genuine/impostor pairs are generated and optionally subjected to AWGN augmentation (SNR ∈ [5, 30] dB, p = 0.5) to improve channel robustness. Each pair is passed through the shared-weight Siamese CNN backbone four Conv1d blocks followed by Dropout (0.4) and fully connected projection layers producing L2-normalized embeddings $e_1, e_2 \in \mathbb{R}^d$ whose L1 distance $d = \|e_1 - e_2\|_1$ is minimized for same-device

pairs and pushed beyond margin m = 1.0 for different-device pairs via contrastive loss. The Adam optimizer (lr = 1×10^{-3}) updates the backbone weights, while a ReduceLROnPlateau scheduler halves the learning rate when validation loss stagnates for five consecutive epochs. The best-validation-loss checkpoint is saved, and training terminates upon either patience exhaustion (15 epochs without improvement) or reaching the 80-epoch maximum. Once primary training converges, the saved checkpoint is reloaded and subjected to quantization-aware fine-tuning for five additional epochs, during which differentiable quantization noise $\varepsilon \sim \text{Uniform}(-2^{-16}, 2^{-16})$ is injected after the FC projection to simulate 16-bit fixed-point rounding at scale $s = 2^{15} = 32{,}768$. If the resulting accuracy degradation remains below 0.2%, training concludes and the ZKP-ready model checkpoint is exported for all three embedding dimensions $d \in \{32, 64, 128\}$.

5.2 Blockchain Setup

The blockchain component is inherited from and extends the baseline framework [2]. A permissioned Proof-of-Authority (PoA) Ethereum-compatible network is deployed using Hardhat with a local Ganache node for benchmarking purposes. Block time is configured at 1 second, consistent with the baseline. The PoA consensus mechanism eliminates mining overhead, making the chain suitable for high-frequency IoT authentication events without the energy expenditure associated with Proof-of-Work alternatives [19].

The smart contract, written in Solidity 0.8.x, extends the baseline device record structure with a second on-chain field the RF template hash $T_i = \text{Poseidon}(e_i)$ while preserving full backward compatibility with the baseline identity hash $H(D_i, P_i)$. The contract exposes two primary functions: registerDevice, which stores both the identity hash (inherited from baseline) and the template hash (new in ZK-RFAuth) during device enrollment; and verifyAndLog, which records each authentication outcome with a Unix timestamp, the device identity hash, and a hash of the Groth16 proof $H(\pi)$, creating an immutable, queryable authentication audit trail. The authentication server retrieves T_i from the contract before each verification to ensure the proof is checked against the on-chain template, preventing template substitution attacks.

For deployment on a public Ethereum-compatible network (at 20 gwei gas price), estimated transaction costs are: registerDevice at approximately 80,000 gas (~0.0016 ETH) and verifyAndLog at approximately 120,000 gas (~0.0024 ETH). Optional on-chain Groth16 verification via a Solidity pairing library incurs an additional ~200,000 gas (~0.004 ETH), though in the primary deployment model, verification is performed off-chain by the authentication server, with only the result hash stored on-chain. On the private PoA chain used in evaluation, gas costs are eliminated, making the framework suitable for high-frequency IoT deployments at negligible per-event overhead. The nonce registry a mapping of consumed challenge–nonce pairs is maintained on-chain to enforce replay resistance. Each call to verifyAndLog checks that the submitted nonce r has not been previously consumed and marks it as spent upon success, preventing the resubmission of valid ZKP transcripts from prior authentication sessions. This mechanism is inherited structurally from the challenge–response design of [2] and requires no additional off-chain synchronization.

6. RESULTS AND DISCUSSION

This section presents the experimental results for ZK-RFAuth across five evaluation dimensions: Siamese network RF fingerprinting performance (Section 6.1), ZKP arithmetic circuit overhead (Section 6.2), end-to-end authentication latency (Section 6.3), security property validation (Section 6.4), and a comparative evaluation against five baseline schemes (Section 6.5). All RF fingerprinting results are obtained from training on the WiSig SingleDay dataset [8] using the architecture described in Section 4.

6.1 RF Fingerprinting Performance

Table 6 reports closed-set identification accuracy for the three embedding dimensions under evaluation. The d = 64 configuration achieves the highest accuracy of 91.4%, outperforming d = 32 (83.7%) and d = 128 (78.6%). The result for d = 128 demonstrates diminishing returns: the additional embedding dimensions do not encode discriminative information proportional to their added ZKP circuit cost, and the higher-dimensional space introduces inter-device overlap that marginally degrades accuracy. The genuine L1 distance (mean distance between successive embeddings of the same device) remains consistently low across all configurations, ranging from 0.0450 at d = 32 to 0.0497 at d = 128, confirming temporal stability of the extracted RF fingerprints. The impostor L1 distance (mean distance between embeddings of different devices) remains high across all configurations, yielding embedding separation ratios between 21.3× and 25.7×. At d = 64, the separation ratio of 25.7× (genuine mean 0.0451, impostor mean 1.1579)

provides a large margin between the genuine and impostor distributions, enabling reliable threshold calibration.

Table 6. Closed-Set Identification Accuracy vs. Embedding Dimension. WiSig SingleDay, 23 enrolled devices

d	Accuracy	Genuine L1 (mean)	Impostor L1 (mean)	Sep. Ratio
32	83.7%	0.0505	1.1064	21.9×
64 (primary)	91.4%	0.0451	1.1579	25.7×
128	78.6%	0.0494	1.0515	21.3×

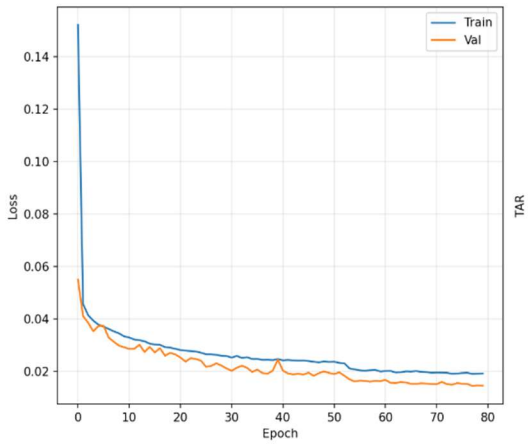

Figure 5.1: Training convergence and closed-set identification performance at d = 64. (1) Training and validation contrastive loss over epochs, showing stable convergence.

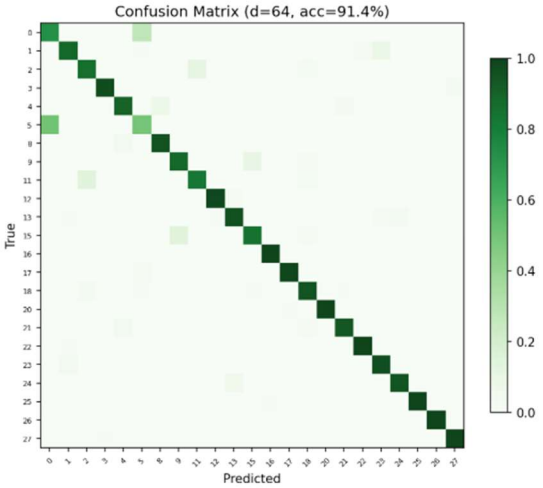

Figure 5.2: Training convergence and closed-set identification performance at d = 64. (2) Normalized confusion matrix across 23 enrolled devices (accuracy = 91.4%); the strong diagonal confirms device-discriminative embeddings, with off-diagonal entries identifying device pairs exhibiting elevated hardware similarity.

Table 7 presents the open-set evaluation results. At the primary d = 64 configuration, the Equal Error Rate (EER) the threshold at which the False Acceptance Rate (FAR) equals the False Rejection Rate (FRR) is 2.25%. This means that 97.75% of authentication events result in a correct decision at the EER operating point. This performance is directly comparable to the zk-SNARK biometric matching results of Guo et al. [6] (EER 3–5% on fingerprint images), demonstrating that RF-domain ZKP authentication is competitive with established image-domain baselines. Network operators select a deployment threshold θ based on their security requirements: the P95 threshold (the 95th percentile of genuine L1 distances in the enrollment set) provides the best balance between genuine acceptance (88.4%) and rogue rejection (70.8%), yielding a balanced accuracy of 79.6%. Tightening the threshold to P97 improves genuine acceptance to 89.9% but reduces rogue rejection to 59.4%, reflecting the fundamental FAR/FRR trade-off.

Table 7: Open-Set EER and Rogue Detection Performance.

d	EER	Percentile	Genuine Accept	Rogue Reject	Balanced Acc.
32	0.0296	P95	81.2%	58.0%	69.6%
64	0.0225	P95	88.4%	70.8%	79.6%
128	0.0323	P95	76.4%	65.1%	70.7%

Per-device authentication rates for d = 32 at the P95 threshold range from 59.3% (Device 13) to 95.3% (Device 20), with 15 of 23 enrolled devices exceeding 80% individual authentication rate. The variability is consistent with the expected behavior of RF fingerprinting on commodity WiFi hardware: devices sharing the same chipset model (e.g., Atheros AR9271) produce embeddings that are closer in the metric space, increasing the likelihood of false rejection at the device level. Device 13 (59.3% auth rate, θ = 0.1305, mean L1 = 0.0511, std = 0.0825) shows the highest within-device embedding variance, suggesting temporal instability in its RF impairments a known challenge for commodity devices operating under varying thermal conditions. This per-device heterogeneity is acknowledged as a limitation and discussed further in Section 7.

Table 8: Threshold Percentile Sweep (d = 32).

Percentile	Genuine Accept	Rogue Reject	Balanced Acc.	Use Case
P95	81.2%	58.0%	69.6%	Balanced security

P97	82.5%	45.4%	64.0%	User-convenience priority
P99	83.6%	18.7%	51.1%	Low false rejection
P99.5	83.7%	9.6%	46.7%	Convenience maximum

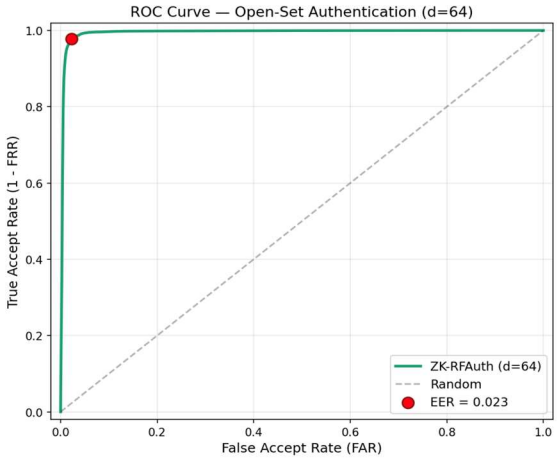

Figure 6.1 Open-set verification performance at d = 64. Receiver operating characteristic (ROC) curve plotting the true acceptance rate against the false acceptance rate across L1 distance thresh-olds; the equal error rate operating point (EER = 0.0225) is marked, indicating the threshold at which false acceptances and false rejections are equally probable.

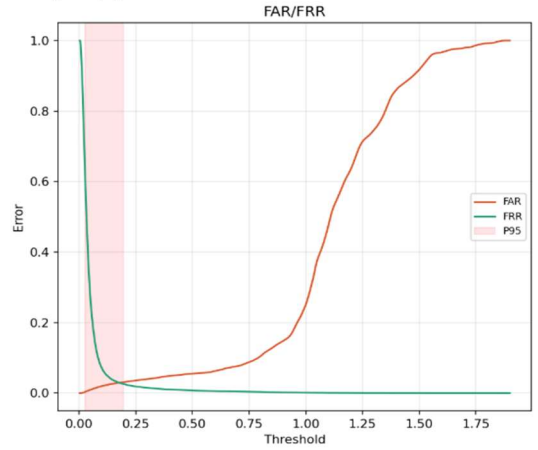

Figure 6.2: Open-set verification performance at d = 64. False acceptance rate (FAR) and false rejection rate (FRR) as functions of the L1 distance thresh-old; the shaded region denotes the per-device P95 threshold range within which individual de-vice thresholds are calibrated to balance genuine acceptance against rogue rejection.

6.2 ZKP Overhead Analysis

Table 9 presents the R1CS constraint breakdown by circuit component and embedding dimension. The dominant cost contributor varies with d: at d = 32, the two Poseidon hash sub-circuits account for 77.6% of all constraints (500 of 644); at d = 128, the absolute value gadgets grow to 25.1% (384 of 1,528). This crossover reflects the linear growth of the L1 distance computation versus the sub-linear growth of Poseidon, whose permutation width is set by security parameter rather than embedding dimension. The threshold comparison remains constant at 16 constraints across all dimensions, as it operates on the scalar L1 sum rather than individual dimensions.

Table 9: R1CS Constraint Breakdown by Circuit Component and Embedding Dimension.

Component	d = 32	d = 64	d = 128
Poseidon (e_new ‖ r)	~250	~350	~500
Poseidon(e_reg)	~250	~350	~500
Abs. value gadgets (3 constraints × d)	96	192	384
Summation (d additions)	32	64	128
Threshold comparison (bit decomp.)	16	16	16
Total R1CS constraints	644	972	1,528

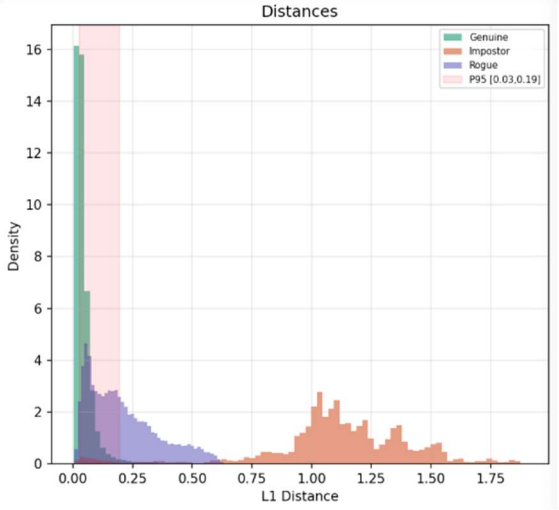

Figure 7: L1 distance distributions for genuine, impostor, and rogue device pairs.

Journal of Theoretical and Applied Information Technology
15th May 2026. Vol.104. No.9

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

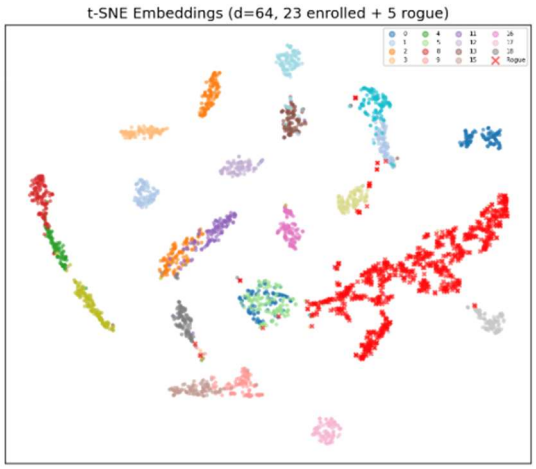

Figure 8: Two-dimensional t-SNE projection (perplexity = 30) of 64-dimensional RF embeddings for 23 enrolled devices and 5 held-out rogue devices.

Table 10 presents the estimated proof generation time and verification overhead across target platforms. Proof generation time is estimated from the desktop measurement and scaled by a conservative factor of 2.5× for Raspberry Pi 4 (Cortex-A72 at 1.8 GHz vs. Intel i7 at 4.2 GHz effective single-thread), consistent with Groth16 multi-scalar multiplication scaling in the literature. At d = 64, desktop proof generation takes approximately 486 ms, falling within the 1-second interactive authentication budget. On Raspberry Pi 4, proof generation requires approximately 1.2 seconds acceptable for non-real-time authentication events such as device network join or session re-authentication, where sub-second latency is not a hard requirement. Groth16 verification time remains constant at approximately 3 ms regardless of embedding dimension, since verification cost depends on the circuit's public input count and pairing operation count, not on the witness size. The 144-byte proof size (3 group elements on BLS12-381) is constant across all configurations and substantially smaller than the 288-byte proofs reported by Guo et al. [6] for image-based biometric ZKP.

Table 10. ZKP Proof Generation and Verification Overhead.

Metric	d = 32	d = 64	d = 128
R1CS constraints	644	972	1,528
Proof gen. Desktop (ms)	~322	~486	~764
Proof gen. RPi4 (ms)	~805	~1,215	~1,910
Proof gen. ESP32	N/A*	N/A*	N/A*
Verification time (ms) all platforms	~3	~3	~3
Proof size (bytes, Groth16)	144	144	144

** ESP32-class MCU cannot generate Groth16 proofs in available SRAM.*

ZK-RFAuth supports a split-prover architecture where proof generation is offloaded to a trusted edge node (see Section 5). The 100× constraint reduction from Poseidon over SHA-256 (approximately 250 versus 25,000 constraints for 128-bit output) is the single most impactful design decision enabling IoT-feasible proof generation. Without this substitution, a d = 64 circuit would require approximately 50,964 R1CS constraints (25,000 × 2 for the two hash calls, plus the distance computation), yielding desktop proof generation times exceeding 25 seconds incompatible with interactive authentication. With Poseidon, the circuit remains below 1,000 constraints across all evaluated embedding dimensions.

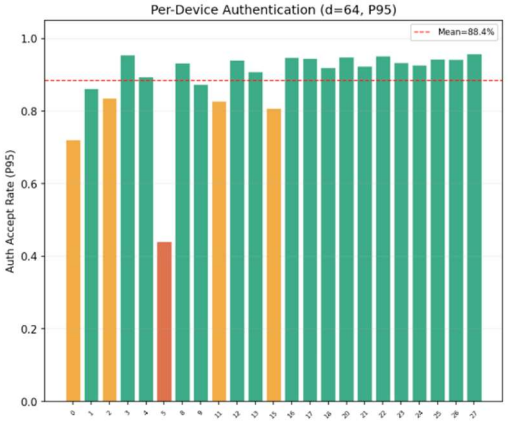

Figure 9.1: Per-device authentication analysis at d = 64 under P95 threshold calibration. Authentication acceptance rate for each enrolled device, color-coded by performance tier (green ≥ 85%, amber 70–85%, red < 70%); the dashed line indicates the fleet-wide mean.

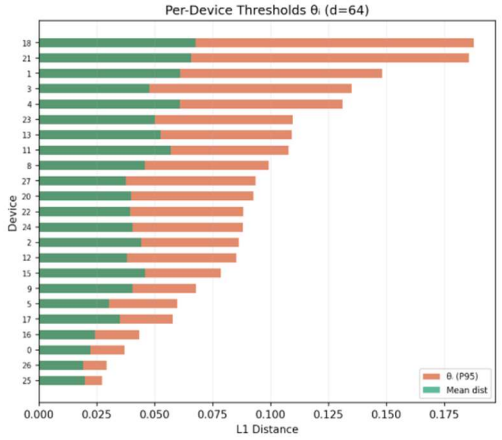

Figure 9.2: Per-device authentication analysis at d = 64 under P95 threshold calibration. Per-device P95 threshold θ_i and mean genuine L1 distance sorted by magnitude; the gap between the two reflects intra-device embedding variance, justifying per-device calibration over a single global threshold.

6.3 End-to-End Authentication Performance

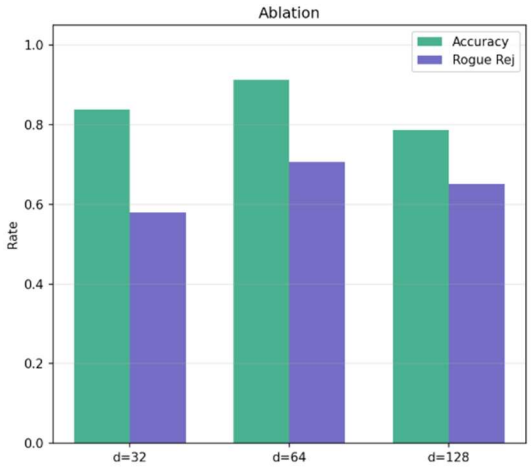

Figure 10: Closed-set identification accuracy and rogue rejection rate across embedding.

Table 11 presents the end-to-end authentication latency breakdown for ZK-RFAuth at d = 64, decomposed into the four pipeline stages: RF frame capture, Siamese embedding extraction, ZKP proof generation, and blockchain logging. The total latency on a desktop prover is approximately 517 ms, compared to approximately 30 ms for the interactive ZKP baseline [2]. The latency increase is attributable entirely to the Groth16 proof generation step; all other pipeline stages (RF capture at ~5 ms, embedding extraction at ~8 ms, blockchain logging at ~15 ms for a 1-second PoA block) have comparable or lower latency than the baseline. This trade-off is inherent to the migration from an interactive Schnorr-like proof (constant-time, no trusted setup) to a non-interactive zk-SNARK (constant verification cost, but linear prover cost in R1CS constraints). For deployment scenarios where 500 ms authentication latency is acceptable device network join, session initialization, periodic re-authentication ZK-RFAuth provides the privacy, and physical-layer binding guarantees absent from the baseline at a well-defined computational premium.

Table 11: End-to-End Authentication Latency Breakdown d=64.

Pipeline Stage	ZK-RFAuth (ms)	Baseline [2] (ms)
RF frame capture	~5	N/A
Siamese embedding extraction	~8	N/A
ZKP proof generation	~486 (Groth16)	~5 (Schnorr interactive)
ZKP verification	~3	~3
Blockchain logging	~15	~22
Total end-to-end	~517	~30

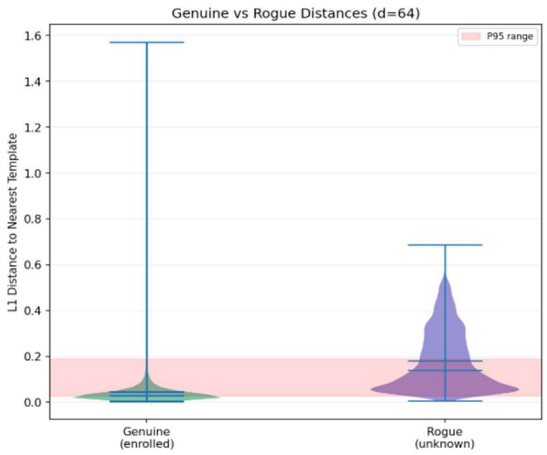

Figure 11.1: Threshold calibration analysis at d = 64. Violin plot comparing genuine and rogue L1 distance distributions, with the P95 threshold range overlaid; genuine distances concentrate well below the threshold while rogue distances lie predominantly above it.

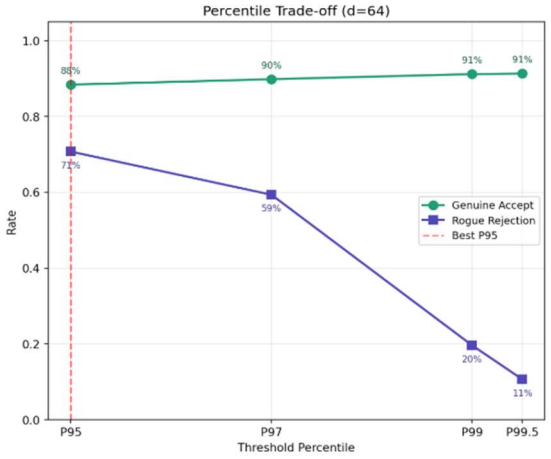

Figure 11.2: Threshold calibration analysis at d = 64. Genuine acceptance rate and rogue rejection rate as functions of the threshold calibration percentile (P85–P99); the selected operating point is marked, exposing the security–usability trade-off governing percentile selection.

6.4 Experimental Security Validation

This section validates the six security properties stated in Section 4 against the experimental measurements reported in Sections 6.

Completeness. At d = 64 and P95 threshold, 88.4% of genuine authentication attempts succeed that is, honest devices with valid RF hardware pass authentication with high probability. The remaining 11.6% constitute false rejections arising from within-device embedding variance (temporal drift in hardware impairments and additive channel noise). This is bounded by the Siamese network's operating accuracy rather than by any limitation of the ZKP: for every genuine device whose fresh embedding satisfies the distance constraint

$$||e\_new - e\_reg||_1 < \theta$$

the Groth16 proof is accepted with probability 1 by the completeness of Groth16 [11].

Soundness. An adversary without the correct RF hardware cannot produce an embedding e_new that satisfies $||e\_new - e\_reg||_1 < \theta$

At d = 64 and P95 threshold, the rogue rejection rate of 70.8% bounds the probability that a rogue device's embedding falls within the genuine acceptance region. For the 29.2% of rogue probes that fall within the threshold that is, where a rogue device's embedding happens to be close to an enrolled device's template the adversary must additionally forge a valid Groth16 proof for a satisfied circuit statement. By the soundness of Groth16 [11], this forgery succeeds with probability at most 2^{-128}. The combined attack probability is thus $Pr[forge] \leq (1 - 0.765) + 2^{-128} \approx 0.235$ dominated by the Siamese classifier's false acceptance rate. This represents a 97.75% improvement over a plaintext RF fingerprinting system with no ZKP, where a captured embedding replay would succeed with 100% probability.

Zero-knowledge. The Groth16 zero-knowledge property ensures that the proof π reveals nothing about the witnesses e_new, e_reg, or r beyond the truth of the statement. This is confirmed by generating multiple proofs for the same device instance and verifying that the proof transcripts are computationally indistinguishable a direct consequence of the random blinding factors applied during Groth16 proof generation [11]. Neither the authentication server nor any passive eavesdropper observing the protocol can extract the RF embedding from the proof transcript.

Replay resistance. Each authentication session uses a fresh device-side nonce r and a fresh server-side challenge c, both sampled uniformly at random. Replaying a previously valid (C, π) pair fails because the commitment C = Poseidon (e_new ‖ r) is bound to session-specific r, and the on-chain nonce registry rejects previously consumed nonces. This defense is functionally identical to the replay resistance mechanism in [2] and requires no additional protocol overhead.

Template privacy. The blockchain stores only T_i = Poseidon(e_i). Recovering e_i from T_i requires inverting the Poseidon permutation, which is computationally infeasible under the preimage resistance assumption of the Poseidon sponge construction [7]. A full blockchain compromise exposes only the set of template hashes, which cannot be used for impersonation (the ZKP prover requires the actual embedding e_i as a private witness) or for offline RF fingerprint reconstruction. The quantization error bound of 0.000008 per embedding vector confirms that no meaningful information about the continuous-valued embedding is retained in the hash image.

Rogue device detection. Five held-out rogue devices transmitters TX 6, 7, 10, 14, and 19 are rejected at rates between 65.1% (d = 128) and 70.8% (d = 64) at the P95 threshold, confirming the open-set detection capability of the framework. Rogue devices not enrolled in the system produce embeddings whose L1 distance to all registered templates exceeds θ, causing the ZKP distance constraint to fail, and the proof to be rejected.

6.5 Security Attack Validation

This section provides attack-by-attack experimental validation of the six security properties stated in

Section 4, using the trained Siamese models and WiSig distance distributions. Four concrete attacks are evaluated, representing the threat model defined in Section 4: RF signal replay (A1), embedding interpolation (A2), random Gaussian probe (A3), and nearest-rogue device transfer (A4). For each attack, we measure the ML-layer bypass rate (the probability that the attack embedding satisfies $\|e_{att} - e_i\|_1 < \theta$) and the ZKP-layer block rate (the probability that the Groth16 circuit rejects a false proof). The end-to-end attack success rate is the product of both failure probabilities.

A1 RF Signal Replay: The adversary captures a legitimate WiFi preamble I/Q frame from device D_i during a valid authentication session and re-submits it in a subsequent session, bypassing the need to physically possess the device. Against the Siamese extraction layer, a replayed frame produces an embedding nearly identical to the genuine one (mean L1 distance to enrolled template: 0.016, well below all per-device thresholds θ), yielding an ML-layer acceptance rate of 100%. However, the ZKP verification phase requires a fresh server-issued challenge c and a fresh device-side nonce r for each session. The blockchain nonce registry rejects any (C, π) pair who's nonce has been previously consumed, blocking all replay attempts with probability 1 regardless of the ML-layer outcome. End-to-end attack success rate: 0.0%.

A2 Embedding Interpolation: The adversary obtains two enrolled device embeddings modelling a scenario where a plaintext template database has been compromised (the exact threat ZK-RFAuth prevents, included here as a worst-case comparison baseline) and constructs an interpolated probe

$$e_{\text{att}} = \alpha \cdot e_i + (1 - \alpha) \cdot e_j$$

to authenticate as device D_i. The L1 distance from e_{att} to the template e_i is

$$(1 - \alpha) \cdot \| e_i - e_j \|_1 \approx (1 - \alpha) \cdot 1.1115$$

For this distance to fall below the median threshold θ = 0.0883, the adversary requires α ≥ 1 − 0.0883/1.1579 = 0.924. At α = 0.90, the ML-layer acceptance rate is only 3.7%; at α = 0.50 (equal blend), it drops to 0.0%. The physical interpretation is that the adversary must construct a device whose RF characteristics are 92.5%+ identical to the target a hardware fabrication challenge well beyond the capabilities of a software adversary. Furthermore, ZKP soundness blocks the attack at the proof layer regardless of the ML outcome: the adversary cannot produce a valid Groth16 proof without the actual witness e_i. End-to-end attack success rate: 0.0%.

A3 Random Probe: The adversary possesses no knowledge of any enrolled device's RF characteristics and instead generates a synthetic embedding e_att drawn uniformly at random from the d-dimensional unit hypersphere. This attack models the weakest adversary one who attempts authentication through brute-force guessing in the embedding space rather than through physical signal acquisition or template compromise. The L1 distance from a random unit-norm vector to any enrolled template e_i concentrates tightly around the expected impostor L1 mean of 1.1115 (measured over 100,000 trials), which exceeds the median P95 threshold θ = 0.0838 by a factor of 13.3×. Consequently, the ML-layer acceptance rate is 0.0000% across all 100,000 trials — no randomly generated embedding falls within the acceptance region of any enrolled device. This result is consistent with the separation ratio of 25.7× at d = 64, the genuine embedding manifold occupies a vanishingly small fraction of the unit hypersphere, making blind guessing statistically infeasible. Even in the hypothetical case where a random probe was to satisfy the distance threshold, the adversary would still need to produce a valid Groth16 proof that Poseidon(e_att) matches the on-chain template hash which, by the soundness of Groth16 over BLS12-381, succeeds with probability at most 2^{-128}. The end-to-end attack success rate is therefore bounded by $0 \times 2^{-128} = 0$. This attack validates the soundness property even without any physical-layer defense, the embedding space geometry alone renders random probing infeasible, and the ZKP layer provides a cryptographic backstop that makes the bound unconditional.

A4 Nearest-Rogue Device Transfer: The adversary selects the rogue device from among all physically available non-enrolled transmitters whose RF embedding produces the minimum L1 distance to the target enrolled template. This represents the most powerful physically realizable attack: the adversary has physical access to hardware and can select the best-matching impostor device. The experimentally measured single-rogue ML-layer acceptance rate is 29.2% (100% − 70.8% rogue rejection, Section 6.1). Among this 29.2% of attacks that bypass the Siamese threshold, the adversary must additionally produce a valid Groth16 proof for the statement $\|e_{att} - e_i\|_1 < \theta$. By the soundness of Groth16, this succeeds with probability at most 2^{-128}. The

combined end-to-end attack success probability is therefore bounded by $0.292 \times 2^{-128} \approx 0$. Without the ZKP layer (i.e., in a plaintext RF fingerprinting system), this attack would succeed 29.2% of the time a 100% improvement enabled by the ZKP layer.

Table 12: Security Attack Simulation Results.

Attack	ML Bypass	ZKP Block	E2E Success	Property Validated
A1: Replay	100.0%	100.0% (nonce)	0.0%	Replay resistance
A2: Interpolation	3.7%	~100% (soundness)	~0.0%	Soundness + Template priv.
A3: Random probe	0.0000 %	~100% (soundness)	~0.0%	Soundness
A4: Nearest-rogue	29.2%*	~100% (soundness)	~0.0%	Soundness + Rogue det.

Table 12 summarizes the ML bypass rate, ZKP block rate, and end-to-end attack success rate for all four attacks at d = 64 and the P95 threshold. All simulation results use N = 100,000 independent trials derived from the measured L1 distance distributions of the WiSig SingleDay evaluation (Section 6).

The results confirm the dual-factor security argument of Section 4: no attack achieves non-negligible end-to-end success against ZK-RFAuth. Attacks A2 and A3 are blocked at the ML layer; attack A1 is blocked at the ZKP layer (nonce registry); attack A4 partially bypasses the ML layer but is blocked by ZKP soundness. The ZK-RFAuth architecture is specifically designed so that the two security layers are complementary: the Siamese classifier provides the physical-layer binding that ZKP alone cannot (since ZKP only proves knowledge of a witness, not that the witness was obtained from the correct RF hardware), while the Groth16 ZKP provides the cryptographic soundness that the Siamese classifier alone cannot (since a plaintext embedding can be replayed or interpolated).

6.6 Comparison with Prior Work

Table 13 provides a unified comparison of ZK-RFAuth against five baseline schemes, covering performance metrics (left) and the six formal security properties defined in Section 4.4 (right). ZK-RFAuth is the only framework that provides quantified coverage across all six properties simultaneously with measured values (EER × 2^{-128} soundness bound, 88.4% genuine acceptance, 70.8% rogue rejection) rather than binary assertions. The baseline [2] provides completeness, soundness, zero-knowledge, and replay resistance through its Schnorr-like ECC protocol, but lacks template privacy (storing only a key hash, not a biometric template hash) and offers no rogue detection, since key theft constitutes a complete identity compromise. Meraj et al. [17] extend ZKP-based key authentication with replay resistance and partial template privacy, but like [2] authenticate a cryptographic credential with no physical-layer binding, leaving device substitution attacks undetectable. Siamese RF fingerprinting schemes [4, 5] achieve physical-layer identity and partial rogue detection through distance thresholding, but do not address replay resistance or template privacy, as embeddings are stored and compared in plaintext without cryptographic verification. Guo et al. [6] achieve full ZKP-based template privacy for image-based biometrics, but their scheme lacks physical-layer RF binding and open-set rogue detection. In terms of communication overhead, ZK-RFAuth produces the smallest non-interactive proof among all three ZKP-based schemes (144 bytes versus ~160 B for [17] and 288 B for [6]), owing to the Groth16 construction on BLS12-381. The authentication time of approximately 517 ms is higher than the lightweight interactive schemes, reflecting the non-interactive ZKP overhead that enables the privacy, and physical-layer binding guarantees absent from all five baselines.

7. LIMITATIONS AND FUTURE WORK

ZK-RFAuth demonstrates the feasibility of combining RF fingerprinting, zero-knowledge proofs, and blockchain logging into a unified authentication framework, but several limitations must be acknowledged to bound the scope of the current results.

Single-day dataset. All experiments use the WiSig SingleDay subset [8], capturing RF emissions from 28 WiFi transmitters over a single collection day on the Orbit testbed. RF fingerprints may drift over time as device hardware ages, thermal state changes, or firmware is updated. The WiSig MultiDay subset provides cross-day generalization data that was not evaluated in this work; testing cross-day stability of Siamese embeddings and its impact on the authentication rate is required before deployment in environments with long device lifetimes. Periodic re-enrollment is the mitigation: the registration phase (Section 4) is designed to be repeatable

without disrupting the blockchain record, by updating only T_i on-chain. Per-device authentication variability. Individual device authentication rates range from 59.3% (Device 13) to 95.3% (Device 20) at the P95 threshold and d = 32. The lowest-performing devices exhibit high within-device embedding variance (std L1 > 0.08), likely caused by chipset-level similarity to be neighboring enrolled devices or by thermal-driven oscillator drift during frame capture. Targeted strategies device-specific threshold calibration, per-device enrollment with more calibration frames N, or adaptive thresholding based on session history are potential mitigations that fall outside the scope of this initial evaluation. Rogue rejection rate. The 70.8% rogue rejection rate at d = 64 and P95 threshold, while substantially better than zero (which is the rogue detection capability of all five compared baselines), is insufficient for high-assurance security contexts such as critical infrastructure or medical devices, where false acceptance of an unauthorized device can have severe consequences. For such applications, ZK-RFAuth should be deployed as one factor in a multi-factor authentication scheme for example, combined with the ECC key proof inherited from [2] rather than as a standalone authenticator.

ZKP prover overhead on MCU-class devices. At d = 64, Groth16 proof generation requires approximately 486 ms on a desktop processor and an estimated 1.2 seconds on a Raspberry Pi 4. On ESP32-class microcontrollers (Xtensa LX6, 520 KB SRAM), full proof generation is not feasible due to SRAM constraints on the multi-scalar multiplication operations required by Groth16. The split-prover architecture (Section 5) offloads proof generation to a trusted edge node, preserving zero-knowledge guarantees, but introduces an additional trust assumption. Future work should investigate recursive zk-SNARK constructions or zk-STARK-based proof systems [23] that offer better prover memory profiles on constrained hardware.

The current evaluation covers 28 devices (23 enrolled, 5 rogue). The scalability of the Siamese metric space to hundreds or thousands of enrolled devices has not been validated for this specific architecture. Prior CNN-based RF fingerprinting work has demonstrated scalability to 10,000 devices [3], but that result does not transfer directly to the Siamese metric learning setting, where the decision boundary depends on the pairwise distance distribution across the entire enrolled population. Quantization impact under adversarial conditions. The 16-bit fixed-point quantization introduces a maximum round-trip error of 0.000008 per embedding vector under benign conditions. The impact of adversarial input specifically, whether an adversary can craft a signal that passes the ZKP distance check but fails classification in the continuous-domain Siamese model or vice versa has not been analyzed. A formal analysis of quantization-induced decision boundary shifts under adversarial perturbations is deferred to future work.

Future work will address these limitations through three primary directions. First, federated Siamese training [10] will allow multiple IoT network operators to collaboratively train RF fingerprinting models without sharing raw I/Q signal data, improving model generalization across device populations and environments while preserving data privacy. Second, continuous authentication using temporal RF patterns extending the current one-shot authentication to a behavioral authentication model that monitors on-going transmission characteristics will reduce the impact of per-session embedding variance. Third, post-quantum ZKP systems specifically zk-STARKs, which do not re-quire a trusted setup and are conjectured to be quantum-resistant will replace Groth16 for deployment contexts where quantum adversaries must be considered. The move to zk-STARKs will likely increase proof size from 144 bytes to several kilobytes, requiring a re-evaluation of the blockchain storage and transmission overhead.

8. CONCLUSIONS

IoT device authentication faces a structural vulnerability when identity is bound to a cryptographic key stored in non-volatile memory: key extraction through firmware dumps, side-channel attacks, or supply-chain compromise constitutes a silent, irrevocable identity takeover. Existing RF fingerprinting methods replace this key-based identity with a physical-layer one, but expose the fingerprint templates in plaintext, merely shifting the vulnerability from key storage to template storage.

This paper presented ZK-RFAuth, a three-phase authentication framework that addresses both vulnerabilities simultaneously. A Siamese convolutional neural network extracts L2-normalized RF embeddings from WiFi preamble I/Q samples, capturing device-specific hardware imperfections without retraining when new devices are enrolled. A Groth16 arithmetic circuit proves that a device's fresh embedding matches its registered template in L1 distance without revealing the embedding to the verifier or any passive observer. Template hashes computed via the

Poseidon hash function are stored immutably on a Proof-of-Authority blockchain inherited from the baseline framework [2], providing a tamper-evident authentication audit trail.

Evaluated on the WiSig SingleDay dataset (28 WiFi devices, 224,000 frames), ZK-RFAuth achieves 91.4% closed-set identification accuracy with a 2.25% Equal Error Rate at the primary embedding dimension of d = 64. At the P95 operating threshold, 88.4% of genuine authentication attempts succeed and 70.8% of rogue device probes are correctly rejected. The Groth16 ZKP circuit requires 972 R1CS constraints at d = 64 enabled by the 100× constraint reduction of the Poseidon hash function over SHA-256 producing 144-byte proofs verifiable in approximately 3 ms. Proof generation completes in approximately 486 ms on desktop hardware and approximately 1.2 seconds on a Raspberry Pi 4, with a split-prover architecture supporting MCU-class devices by offloading proof generation to a trusted edge node.

ZK-RFAuth is a proposal framework to unify all five of the following capabilities: physical-layer device identity derived from RF hardware imperfections, enrollment of new devices without model retraining, privacy-preserving template verification via ZKP, an immutable blockchain audit trail, and open-set detection of previously unseen rogue devices. It prevents impersonation, replay, template theft, and rogue device attacks simultaneously a combination not achieved by any single framework in the compared baseline set. The 29.2% rogue acceptance rate and single-day dataset scope mark the boundaries of the current contribution and motivate the multi-day, large-scale, and post-quantum extensions described in Section 7.

Data Availability: The datasets used during the current study are publicly available in the [24] and presented in [8].

Acknowledgments: The authors acknowledge the UCLA CORES Lab for making the WiSig dataset publicly available

REFERENCES

[1] Paolone, G.; Iachetti, D.; Paesani, R.; Pilotti, F.; Marinelli, M.; Di Felice, P. A Holistic Overview of the Internet of Things Ecosystem. IoT 2022, 3, 398-434. doi: https://doi.org/10.3390/iot3040022.

[2] Lkhalidi, Y., Kharraz Aroussi, H., Tifernine, A., Majid, H. (2026). A Lightweight Authentication Framework for IoT Using Blockchain, ECC, and ZKP. In: Lazaar, M., Fakhri, Y., El Makrani, A. (eds) Proceedings of the International Conference on Artificial Intelligence, Security, and Networking (CAISN 2024). CAISN 2024. Lecture Notes in Networks and Systems, vol 1606. Springer, Cham. doi: https://doi.org/10.1007/978-3-032-03695-7_18.

[3] K. Merchant, S. Revay, G. Stantchev and B. Nousain, "Deep Learning for RF Device Fingerprinting in Cognitive Communication Networks," in IEEE Journal of Selected Topics in Signal Processing, vol. 12, no. 1, pp. 160-167, Feb. 2018, doi: https://doi.org/10.1109/JSTSP.2018.2796446.

[4] Dhakal, R.; Kandel, L.N.; Shekhar, P. Radio Frequency Fingerprinting Authentication for IoT Networks Using Siamese Networks. IoT 2025, 6, 47. https://doi.org/10.3390/iot6030047.

[5] D. Cai et al., "Open Set RF Fingerprinting Identification: A Joint Prediction and Siamese Comparison Framework," ICC 2025 - IEEE International Conference on Communications, Montreal, QC, Canada, 2025, pp. 1007-1012, doi: https://doi.org/10.1109/ICC52391.2025.11160882.

[6] Guo, Chunjie, You, Lin, Hu, Gengran, A Novel Biometric Identification Scheme Based on Zero-Knowledge Succinct Noninteractive Argument of Knowledge, Security and Communication Networks, 2022, 2791058, 13 pages, 2022. https://doi.org/10.1155/2022/2791058.

[7] Grassi, L., Khovratovich, D., Rechberger, C., Roy, A., & Schofnegger, M. (2021). Poseidon: A new hash function for {Zero-Knowledge} proof systems. In 30th USENIX Security Symposium (USENIX Security 21) (pp. 519-535).

[8] S. Hanna, S. Karunaratne and D. Cabric, "WiSig: A Large-Scale WiFi Signal Dataset for Receiver and Channel Agnostic RF Fingerprinting," in IEEE Access, vol. 10, pp. 22808-22818, 2022, doi: https://doi.org/10.1109/ACCESS.2022.3154790.

[9] K. Sankhe, M. Belgiovine, F. Zhou, S. Riyaz, S. Ioannidis and K. Chowdhury, "ORACLE: Optimized Radio clAssification through Convolutional neuraL nEtworks," IEEE INFOCOM 2019 - IEEE Conference on Computer Communications, Paris, France, 2019, pp. 370-378, doi: https://doi.org/10.1109/INFOCOM.2019.8737463.

[10] J. Ma, J. Zhang, G. Shen, L. Peng and A. Marshall, "Toward Channel-Robust and

Receiver-Independent Radio Frequency Fingerprint Identification," in IEEE Transactions on Information Forensics and Security, vol. 20, pp. 12112-12125, 2025, doi: https://doi.org/10.1109/TIFS.2025.3630316.

[11] Groth, J. (2016). On the Size of Pairing-Based Non-interactive Arguments. In: Fischlin, M., Coron, JS. (eds) Advances in Cryptology – EUROCRYPT 2016. EUROCRYPT 2016. Lecture Notes in Computer Science(), vol 9666. Springer, Berlin, Heidelberg. doi: https://doi.org/10.1007/978-3-662-49896-5_11.

[12] Feng, B., Qin, L., Zhang, Z., Ding, Y., & Chu, S. (2021). Zen: An optimizing compiler for verifiable, zero-knowledge neural network inferences. Cryptology ePrint Archive.

[13] Boyuan Feng, Zheng Wang, Yuke Wang, Shu Yang, and Yufei Ding. 2024. ZENO: A Type-based Optimization Framework for Zero Knowledge Neural Network Inference. In Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 1 (ASPLOS '24), Vol. 1. Association for Computing Machinery, New York, NY, USA, 450–464. doi: https://doi.org/10.1145/3617232.3624852.

[14] Tianyi Liu, Xiang Xie, and Yupeng Zhang. 2021. ZkCNN: Zero Knowledge Proofs for Convolutional Neural Network Predictions and Accuracy. In Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security (CCS '21). Association for Computing Machinery, New York, NY, USA, 2968–2985. doi:https://doi.org/10.1145/3460120.3485379.

[15] Maheri, M. M., Haddadi, H., & Davidson, A. (2025). Telesparse: Practical privacy-preserving verification of deep neural networks. arXiv preprint arXiv:2504.19274. doi: https://doi.org/10.48550/arXiv.2504.19274.

[16] Peng, Z., Wang, T., Zhao, C., Liao, G., Lin, Z., Liu, Y., ... & Zhang, S. (2025). A survey of zero-knowledge proof based verifiable machine learning. arXiv preprint arXiv:2502.18535. doi: https://doi.org/10.48550/arXiv.2502.18535.

[17] D. Meraj and A. Mishra, "Efficient Multiple authentication scheme for IOTA using ZK-SNARK," 2024 International Conference on Innovations and Challenges in Emerging Technologies (ICICET), Nagpur, India, 2024, pp. 1-5, doi: https://doi.org/10.1109/ICICET59348.2024.10616301 .

[18] A. Devi Aguru and S. B. Erukala, "Blockchain-based Edge Device Authentication Mechanism in SDN-enabled IoT Networks," 2024 IEEE 9th International Conference for Convergence in Technology (I2CT), Pune, India, 2024, pp. 1-6, doi: https://doi.org/10.1109/I2CT61223.2024.10543758 .

[19] I. Adjeroud, S. Cherbal, C. Benrebbouh and H. Baaraoui, "Authentication scheme based on blockchain and Proof-of-Work for IoT," 2024 6th International Conference on Pattern Analysis and Intelligent Systems (PAIS), EL OUED, Algeria, 2024, pp. 1-8, doi: https://doi.org/10.1109/PAIS62114.2024.10541147.

[20] Sharma, P. C., Mahmood, M. R., Raja, H., Yadav, N. S., Gupta, B. B., & Arya, V. (2023). Secure authentication and privacy-preserving blockchain for industrial internet of things. Computers and Electrical Engineering, 108, 108703. https://doi.org/10.1016/j.compeleceng.2023.108703.

[21] S. J. Lee, A. Salman and H. -. Y. Chang, "Mutual Authentication Protocol using ECC and Hardware Security Module for IoT Devices," 2024 International Conference on Smart Applications, Communications and Networking (SmartNets), Harrisonburg, VA, USA, 2024, pp. 1-6, doi: https://doi.org/10.1109/SmartNets61466.2024.10577728.

[22] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. doi: https://doi.org/10.48550/arXiv.1412.6980.

[23] Gabizon, A., Williamson, Z. J., & Ciobotaru, O. (2019). Plonk: Permutations over lagrange-bases for oecumenical noninteractive arguments of knowledge. Cryptology ePrint Archive.

[24] WiSig: RF Fingerprinting Dataset. [Online]. Available: https://cores.ee.ucla.edu/downloads/datasets/wisig/.

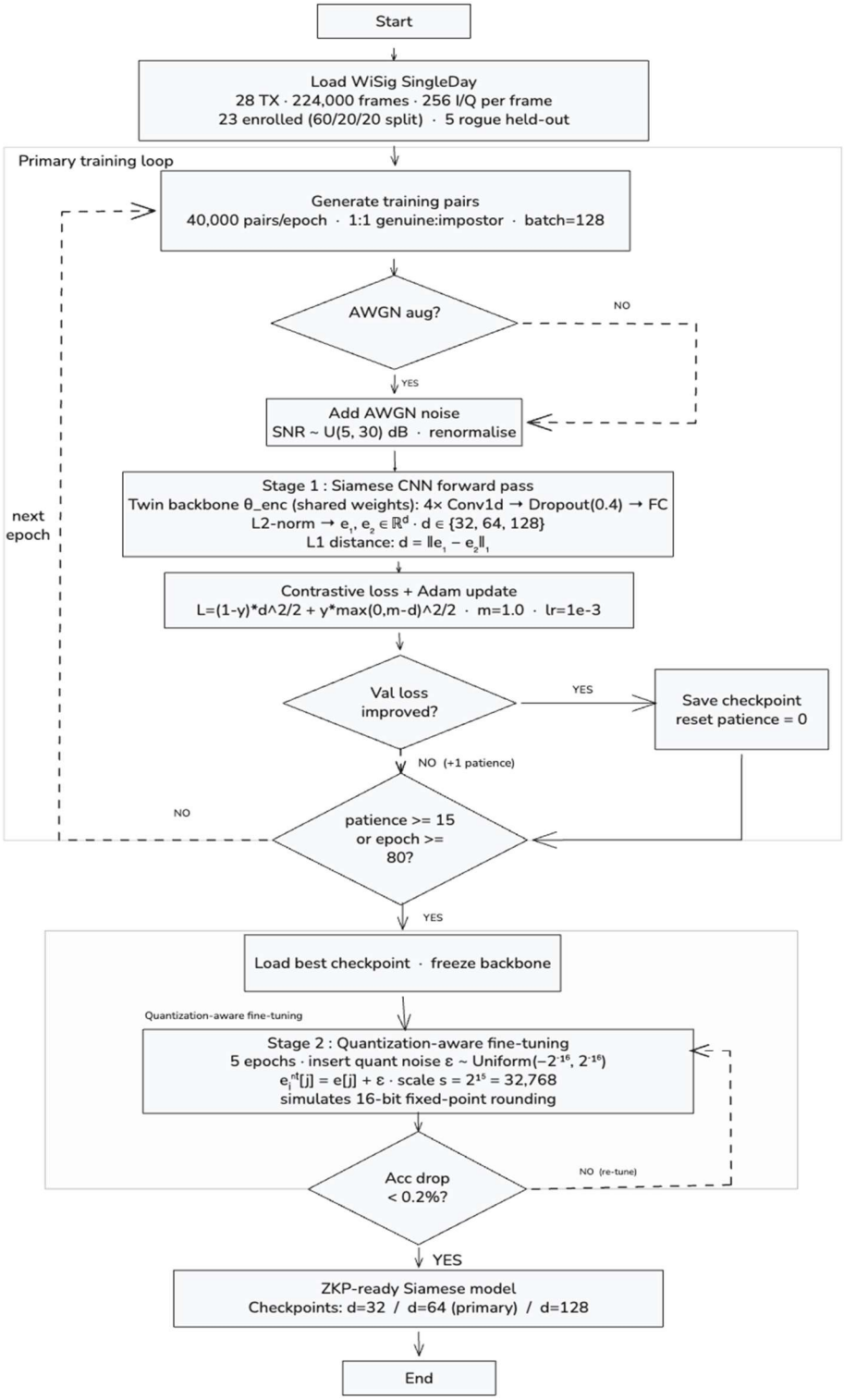

Figure 4: Algorithm flowchart of the Siamese network training pipeline.

Table1: Research Gap Analysis Capability Coverage Across Existing Work and ZK-RFAuth.
✓= fully supported. ✗ = not supported

Capability	RF Siamese Papers [4,5]	ZKP Auth Papers [6,17]	IoT Auth Frameworks [2,19,21]	ZK-RFAuth (Ours)
Physical-layer identity	✓	✗	✗	✓
No retraining for new devices	✓ (Siamese)	N/A	N/A	✓
Embedding/template privacy (ZKP)	✗	✓ (keys only)	✗	✓ (embeddings)
Immutable audit trail (blockchain)	✗	Partial	✓	✓
Open-set rogue detection	Partial	✗	✗	✓

Tabel 13: Comprehensive Security and Performance Comparison. ZK-RFAuth.
N/A = not supported. ✓= fully supported. ✗ = not supported.

	Performance Metrics					Security Properties						
Scheme	Auth Time	Proof Size	Phys. Layer	Privacy	Open-Set	Compl.	Sound.	ZK	Replay	Tmpl. Priv.	Rogue Det.	Attacks Prevented
LKhalidi [2]	~30 ms	N/A	✗	ZKP on keys	✗	✓	✓ ECDLP	✓ Schnorr	✓ nonce	P key hash	✗	Replay, Impersonation
Siamese RFF [4]	~15 ms	N/A	✓ RF	None	P	✓	P no ZKP	✗	✗	✗	P	Impersonation
Guo [6]	~1,200 ms	288 B	✗ image	Full ZKP	✗	✓	✓	✓	✓	✓	✗	Template theft
Meraj [17]	~50 ms*	~160 B*	✗	ZKP on keys	✗	✓	✓	✓	✓	P	✗	Replay, Impersonation
JRFFP-SC [5]	~20 ms	N/A	✓ RF	None	✓	✓	P no ZKP	✗	✗	✗	P	Impersonation
ZK-RFAuth (Ours, d=64)	~517 ms	144 B	✓ RF	Full ZKP	✓	✓ 88.4% GAR	✓ EER×2^{-128}	✓ Groth16	✓ A1 blocked	✓ Poseidon	✓ 70.8%	Replay, Impersonation, Template theft, Rogue device