



PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH
UNIVERSITY OF AIN TEMOUCHENT BELHADJ BOUCHAIB
FACULTY OF SCIENCE AND TECHNOLOGY
DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE

MASTER'S THESIS

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF COMPUTER SCIENCE

SPECIALTY: ARTIFICIAL INTELLIGENCE AND CYBER SECURITY

Decentralized Authentication based on Artificial Intelligence and Blockchain for Internet of Things Environments

EXAMINATION COMMITTEE

DR. FETHI IMAD BENARIBI	PRESIDENT	UNIV. OF AIN TEMOUCHENT
DR. HICHEM BOUCHAKOUR ERRAHMANI	EXAMINER	UNIV. OF AIN TEMOUCHENT
DR. MANSOUR MEDEDJEL	SUPERVISOR	UNIV. OF AIN TEMOUCHENT

PRESENTED BY

MISS BENADLA CHAIMAA
MISS BACHIR NOUR EL IMANE CHAIMAA

DEFENSE DATE

JUNE 27, 2026

Academic Year 2025 / 2026

Dedication



As a sign of love, respect, and gratitude,
for all the sacrifices you have made
for our education and well being.

No dedication could ever express our deep love,
our esteem, our heartfelt gratitude,
our intimate attachment, and our profound affection.

To our parents — first and always.

To our sisters and brothers, our strength and refuge.

To our cousins, companions of our happiest days.

To our teachers.

We dedicate this modest work.

Acknowledgements

Alhamdulillah, all praise and thanks are due to **Allah** for His blessings and for granting us the strength to accomplish this work.

Our sincere gratitude goes to our supervisor, **Dr. Mansour Mededjel**, for his valuable guidance and continuous support.

We dedicate our deepest appreciation to our beloved parents, **Boucif, Kadour** and Mothers **Trache Irchad, Sidi yakoub Fatiha**, whose love, sacrifices, and encouragement have always inspired us.

We also extend our heartfelt thanks to our brother **Abd el samad** and sister **Imane**, for their support and motivation throughout our journey.

Special thanks to our best friends for their friendship and encouragement.

To all those who supported us, thank you from the bottom of our hearts.

Table of Contents

List of Figures	6
List of Tables	7
General Introduction	8
1. Context	8
2. Problem statement	9
3. The proposed approach	9
4. Objectives and thesis organization	10
IoT and Security Challenges	12
1 Introduction and Fundamental Concepts of IoT	12
2 IoT Architecture	13
2.1 Taxonomy of IoT Architectural Models	13
3 Threats and attacks in IoT ecosystems	17
4 Key Challenges in IoT Deployments	18
4.1 Security and privacy challenges	18
4.2 Technical and operational complexity	18
4.3 Connectivity and energy dependency	18
4.4 Protocol Heterogeneity and Integration	19
4.5 Vulnerability to Device Hijacking	19
5 Conclusion	19
Technologies for IoT System Security	20
1 Introduction	20
2 Authentication and Access Control Concepts	21
2.1 Traditional Authentication and Access Control Models and Their Limita- tions for IoT	21
2.2 The Shift Towards Next-Generation Paradigms	21

3	Distributed Ledger Technologies and Blockchain in IoT	22
3.1	Blockchain and Smart Contract Fundamentals	22
3.2	Consensus Mechanisms in IoT Systems	23
4	Blockchain-Based Authentication	24
5	Artificial Intelligence for Anomaly Detection in IoT	26
5.1	Taxonomy of AI-Driven Detection Techniques	27
5.2	How to Link AI with Blockchain	28
6	Analysis of Existing Work	29
6.1	Lightweight Cryptography and Authentication Mechanisms	29
6.2	Blockchain-Based Decentralized Authentication and Privacy	30
6.3	Machine Learning for Intrusion Detection and Vulnerability Analysis	30
6.4	Synthesis and Positioning	31
7	Conclusion	32
Proposed Architecture, Implementation and Evaluation		33
1	Introduction	33
2	Proposed System Architecture	34
2.1	IoT Devices Layer	34
2.2	Backend API Layer	34
2.3	Blockchain Layer	35
2.4	AI Detection Layer	35
2.5	Inter-Layer Communication	35
3	Core Technology Choices	35
3.1	ECDSA for Lightweight IoT Authentication	35
3.2	Zero-Knowledge Proofs for Identity Confidentiality	36
3.3	BiLSTM Autoencoder for Anomaly Detection	37
4	System Implementation and Security Mechanisms	38
4.1	Cryptographic Layer (Layers 1 & 2)	38
4.2	Blockchain and Smart Contracts (Layer 3)	39
4.3	Deep Learning Engine Implementation (Layer 4)	40
5	System Integration, Security Scenarios and Interface	43
5.1	Automated Smart Contract Revocation Mechanism	43
5.2	Simulated Real-Time Attack Scenarios	44
5.3	Monitoring Dashboard Interface	48
6	Performance Evaluation and Discussion	50

6.1	Experimental Environment and Deployment Setup	50
6.2	Authentication, ZKP Circuit and Throughput Metrics	50
6.3	Blockchain Transaction and Gas Consumption Analysis	54
6.4	AI Model Anomaly Detection Accuracy	55
6.5	Discussion and Limitations	58
7	Conclusion	58
General Conclusion		60
References		62
Chapitre A: Source Code — Technical Implementations		69
1	ECDSA — Signature Generation	69
2	ECDSA — Verification	69
3	ZKP — Circuit	70
4	ZKP — Generation and Verification	70
5	Blockchain Recording	70
6	Device Registration	71
7	ZKP Generation Handler	72
8	Authenticated Message Handler	73

List of Figures

1	The three-layer IoT model (adapted from[46])	14
2	The five-layer IoT model (adapted from[33])	15
3	Cloud-Fog-Edge IoT Architecture (adapted from [1])	16
4	Blockchain Structure	23
5	Blockchain-based authentication process	26
6	Blockchain-based authentication VS traditional centralized authentication(adapted from [3])	26
7	Four-Layer IoT System Architecture	34
8	Architecture of the BiLSTM Autoencoder Model	42
9	CHAIMAN IoT dashboard — AI Detection panel during the three-profile attack simulation.	46
10	Dashboard Overview	48
11	Device Management	48
12	Transactions Dashboard	49
13	Security Dashboard	49
14	AI Detection Interface	50
15	Registration latency breakdown: HTTP (T1) vs. blockchain confirmation (T2) .	51
16	Authentication latency breakdown across five deployment phases	52
17	Throughput scaling from 0.65 to 5.43 req/s across deployment phases	54
18	Confusion Matrix of the BiLSTM Autoencoder on Test Set	55
19	Training and Validation Loss Curves (convergence at epoch 127)	56
20	ROC Curve: AUC = 0.9375	56

List of Tables

1	Summary of Existing Works on IoT Security Technologies (2019–2024)	31
2	Recent Studies Validating ECDSA for IoT Authentication	36
3	Recent Studies Validating ZKP for IoT Authentication	37
4	Security Comparison: ECDSA Alone versus ECDSA + ZKP	38
5	Structure of a 65-byte ECDSA Signature (secp256k1)	38
6	Structure of a 192-byte Groth16 Proof on BN254 (uncompressed)	39
7	Feature Set Description (14 Features)	41
8	Attack Profiles and Targeted BiLSTM Severity Levels	45
9	DDoS Attack Traffic Characteristics vs. Normal Ranges	45
10	Integrated Security Mechanisms	47
11	Registration Performance by Phase	51
12	Authentication Performance Breakdown (ms, mean values)	52
13	ZKP Circuit Performance	53
14	Authentication Throughput	53
15	Gas Consumption (units)	54
16	BiLSTM Autoencoder Performance Metrics (Test Set)	57
17	Comparison with Isolation Forest Baseline (Test Set)	57

CHAPTER 1

GENERAL INTRODUCTION

1. Context

The Internet of Things (IoT) has changed the way people interact with technology in everyday life. In simple terms, IoT refers to a large network of connected physical and virtual devices from basic sensors in smart homes to complex industrial machines that communicate over the Internet to collect and exchange data [1]. The power of the IoT lies in the fact that these devices can often operate with minimal human intervention. They can sense information from their environment, process it, and react automatically in real time. Because of this, IoT systems are more efficient than traditional approaches and can operate in environments that are difficult or unsafe for humans [46].

Today, the IoT is used in many different areas. In healthcare, for example, wearable devices continuously track patients' health conditions. In transportation, connected systems help improve traffic flow and reduce accidents. In agriculture sensors are used to monitor soil conditions and automate irrigation. Even in everyday homes smart devices improve comfort and energy efficiency. At the same time IoT systems generate large volumes of frequent, small data streams and rely on a massive number of connected devices. It is expected that the number of IoT devices will exceed 30 billion by 2030 [11]. While this growth brings many benefits, it also increases security risks, as every connected device is potentially vulnerable to attack.

Securing IoT systems is not simple, primarily due to their layered structure. Whether we

consider the basic three-layer model (perception, network, and application) or more advanced architectures, each layer introduces its own security challenges [1]. The perception layer includes sensors and RFID tags which are usually resource limited and can be physically attacked or manipulated. The network layer relies on wireless technologies such as WiFi, Bluetooth, or Zigbee, which are exposed to attacks like eavesdropping, spoofing, or denial of service. The application layer deals with user services and therefore raises issues related to privacy, authentication and application-level attacks [1, 25]. Ensuring confidentiality, integrity and availability across all these layers remains one of the biggest challenges in IoT security.

2. Problem statement

Although IoT systems are growing rapidly, their security mechanisms are not evolving at the same pace. Most existing authentication solutions rely on a centralized architecture in which a single server manages identities, verifies users, and controls access. While this approach can work for small systems, it becomes problematic in large-scale IoT environments.

One of the main issues is the single point of failure (SPOF). If the central server fails due to an attack or technical problem, the entire network may be affected, causing devices to lose connectivity [53]. Another major problem is scalability. When a large number of devices attempt to authenticate simultaneously, the central system can easily become overloaded, causing delays or even system failure [27]. In addition, centralized systems require full trust in one authority, which increases the risk of insider attacks or compromise [53].

Additionally, IoT devices have significant limitations in terms of processing power, memory, and energy. Traditional cryptographic methods, such as RSA with large key sizes are often too heavy for such constrained devices [14]. The IoT ecosystem is also very diverse, consisting of different manufacturers, protocols, and hardware capabilities [1]. These limitations, combined with the dynamic nature of IoT networks where devices constantly join and leave, show that a more flexible and decentralized approach is needed.

3. The proposed approach

To address the aforementioned challenges, this thesis proposes a hybrid approach combining blockchain technology and AI (Artificial Intelligence).

Blockchain technology offers a decentralized way to manage trust. Essentially, it is a distributed and immutable ledger shared across multiple nodes in a network. Rather than relying on a central authority, all participants maintain the same copy of the data, improving reliability

and removing the single point of failure. Once data is recorded on the blockchain, it cannot easily be changed or deleted, ensuring strong integrity and transparency [57].

This work uses blockchain technology to create a decentralized authentication system in which IoT devices register their identities directly on the blockchain. The network then verifies authentication through cryptographic proofs. Smart contracts are used to automate key processes, such as registration, authentication checks, and revocation of compromised devices [31, 10]. The Proof of Authority (PoA) consensus mechanism is used to keep the system lightweight and efficient since it offers fast confirmation times and low energy consumption, which fit IoT environments well [60].

However, authentication alone is not enough. A device may be legitimate initially and become compromised later. For this reason, the proposed system includes an AI component based on a BiLSTM (Bidirectional Long Short-Term Memory) autoencoder. This model continuously analyzes network traffic and learns what normal behavior looks like for IoT devices. When unusual patterns appear, such as abnormal traffic volume or suspicious communication behavior, the system raises an alert and triggers an automatic response [43, 26].

Through smart contracts, the system can then revoke or block the affected device without human intervention. In this way, blockchain ensures secure identity management, AI detects abnormal behavior in real time, and smart contracts execute automatic responses.

Several recent studies have examined the integration of blockchain and AI for IoT security [45, 16]. This thesis is part of a larger research project focused on developing an integrated system of these technologies through an IoT environment simulation.

4. Objectives and thesis organization

This thesis's main objective is to design and implement a decentralized IoT authentication framework that combines blockchain-based identity management with AI-based anomaly detection. Leveraging this dual approach aims to significantly enhance the security, scalability, and real-time response capabilities of constrained IoT environments.

To achieve this, the specific objectives are as follows:

- Formulate a lightweight authentication scheme based on ECDSA and zero-knowledge proofs (ZKP) that can operate seamlessly on resource-constrained IoT devices.
- Deploy a Proof of Authority (PoA) blockchain with smart contracts to fully decentralize device registration, authentication, and revocation.

- Leverage a BiLSTM autoencoder to learn normal IoT traffic patterns and detect security anomalies in real time.
- Integrate the AI-driven detection system with blockchain smart contracts to trigger automated mitigation actions whenever a threat is detected.

The remainder of this thesis is organized into three main chapters:

- **Chapter 2: The Internet of Things (IoT) and Security Challenges.** This chapter introduces fundamental IoT concepts and architectural frameworks, analyzing critical security vulnerabilities across different layers. It also highlights the operational and structural limitations of traditional centralized security models in modern deployments.
- **Chapter 3: Technologies for IoT System Security.** This chapter provides the technical background for the work, covering authentication schemes, blockchain fundamentals, consensus mechanisms, smart contracts, and AI-based anomaly detection models. The chapter also provides a related work review.
- **Chapter 4: Design, Implementation, and Evaluation.** This chapter details the architectural design and implementation of the proposed framework, including the IoT simulation environment, blockchain setup, cryptographic layer, and AI model integration. The chapter also presents experiments, results, and evaluations based on attack scenarios and monitoring tools.

CHAPTER 2

Internet of Things (IoT) and Security Challenges

1 Introduction and Fundamental Concepts of IoT

The IoT is defined as a network of interconnected physical and virtual objects that communicate with each other via wireless protocols and global network infrastructures. These objects can capture and transfer data with a limited power supply. This functionality allows them to analyze and process data, and perform actions without human intervention, except for configuration, monitoring, and connection to the network. Furthermore, embedded intelligence ensures that these objects can perform operations faster, more consistently, and with greater reliability than humans. This technology is already widely adopted in critical areas such as healthcare, transportation, and industry. Crucially, IoT technology differs from traditional computing in that data packets are small but transmitted at a much higher frequency. Consequently, as the number of deployed nodes increases exponentially, the complexity of managing and securing these infrastructures grows significantly [46].

2 IoT Architecture

Currently, there is no universally accepted consensus in the scientific community on a standardized IoT architecture. This has led to numerous models being proposed in the literature [1, 35]. To make sense of this diverse landscape, these frameworks are generally grouped into three main categories recognized by both academia and industry [46, 37].

The first of these is the classic three-layer architecture, a foundational framework that divides the system into three basic layers: perception, networking and application. To handle more complex data processing, this model is often expanded into a five-layer architecture by adding a dedicated middleware layer. Alternatively, modern industrial deployments frequently lean toward decentralized Cloud and Edge-Fog models to distribute computing power. However, because end devices are highly resource-constrained, hosting heavy security mechanisms directly on them remains unfeasible. Understanding these different layers is crucial to explaining how data flows from these limited nodes up to a centralized security core, where real-time threat validation actually takes place.

2.1 Taxonomy of IoT Architectural Models

This sub-section provides an overview of the three main architectural frameworks, each representing a different approach to balancing computational constraints with network efficiency.

A) Three-Layer Architecture

The three-layer model provides a baseline conceptual framework for the IoT by breaking down the architecture into its essential components. It organizes data flow linearly from physical capture to ultimate consumption through three distinct levels: the *perception layer* (sensors and actuators), the *network layer* (data transmission), and the *application layer* (user services). As illustrated in Figure 1, this model depicts a highly centralized flow, where all data collected at the bottom must travel directly to the top for processing [1, 46].

Perception layer: often referred to as the sensor layer, which functions analogously to human sensory organs by interacting directly with the physical world. Its primary task is to identify physical objects and collect real-time data from them. Depending on the application's specific requirements, various data collection technologies can be deployed at this level, including Radio Frequency Identification (RFID) tags, 2D barcodes, and diverse specialized sensors. The information captured by these devices generally encompasses environmental factors, spatial location, atmospheric changes, motion, and structural vibrations. However, as this layer is at the

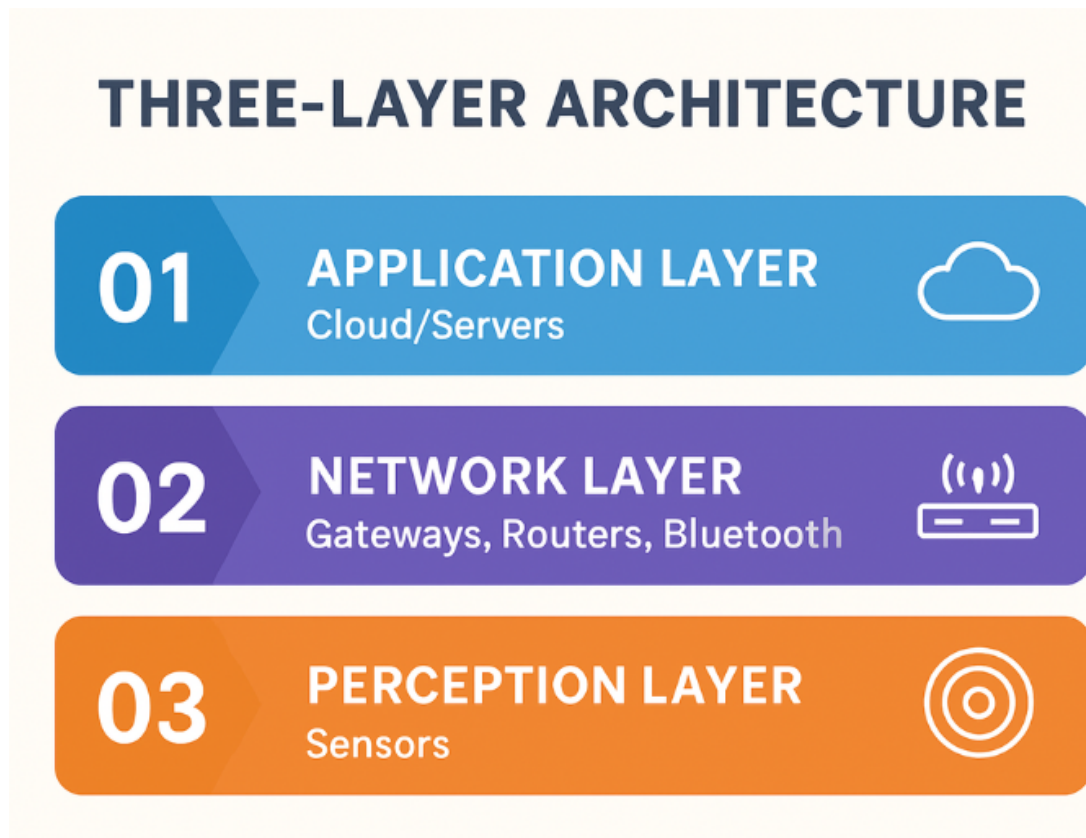


Figure 1: The three-layer IoT model (adapted from[46])

absolute perimeter of the architecture, it represents a prime target for malicious actors. Attackers often exploit the physical vulnerability of these nodes by attempting to compromise them or replace legitimate sensors with rogue devices to inject falsified data into the network.

Network layer: also known as the transmission layer, acts as a critical bridge connecting the physical perception domain to the upper application services. Its core responsibility is to securely transport and route the information collected by sensors across the ecosystem. This communication infrastructure relies on both wired and wireless transmission mediums to interconnect smart objects, routing equipment, and heterogeneous networks. However, because it acts as the central highway for data in transit, this layer is highly vulnerable to sophisticated network attacks. It presents significant security challenges, particularly regarding data integrity and node authentication, as malicious actors often attempt to intercept, alter, or spoof the information flowing through the network channels.

Application layer: at the top of this model, the application layer, which encompasses all the diverse domains where IoT technology is deployed to provide user-centric services. This layer is responsible for translating the raw sensor data into actionable operations tailored to specific sectors, such as smart homes, smart cities, and healthcare systems. As the nature of these services depends entirely on the unique data gathered at the perimeter, the application layer

must remain highly adaptable. However, as the ultimate interface between the infrastructure and the end-user, it faces critical software vulnerabilities, with security being its most significant challenge. Safeguarding this layer is essential, as any compromise here could endanger user privacy, system control, and data validity across the entire network.

B) Five-Layer Architecture

To handle more complex demands relating to the processing and storage of data, researchers expanded the classic three-layer structure into a more detailed five-layer architecture [46]. This model further breaks down the main steps by adding two key layers: the *processing (or middleware) layer*, which manages databases, data filtering and basic analytics; and the *business layer*, which oversees the system's overall logic, privacy policies and applications. As shown in Figure 2, this model creates clear software boundaries. By mapping out these extra checkpoints, the model makes it much easier to determine exactly where security protocols and data encryption should be implemented.

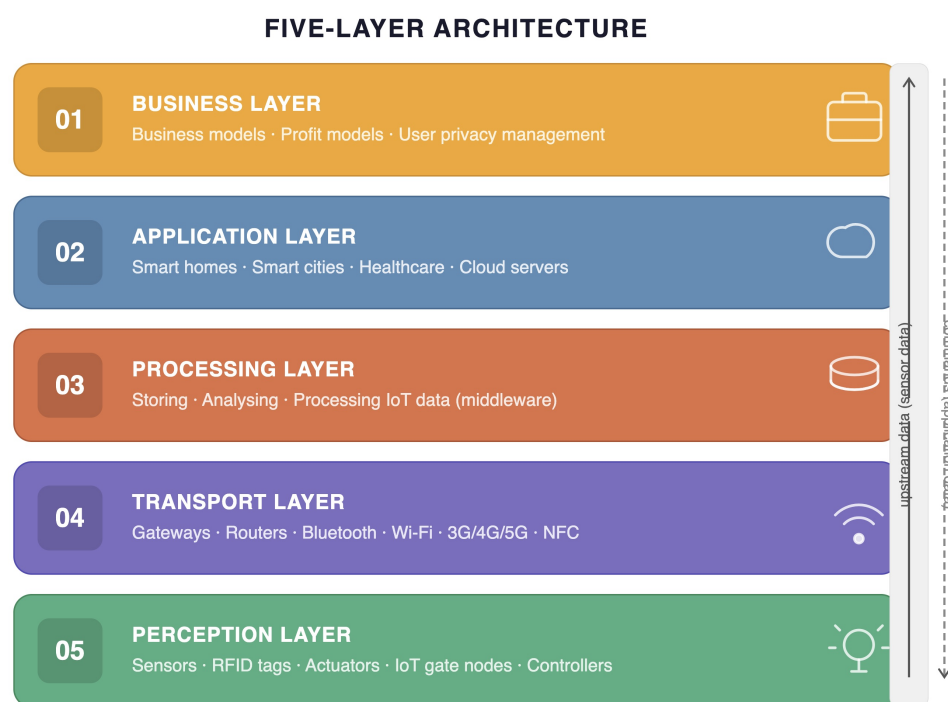


Figure 2: The five-layer IoT model (adapted from[33])

The **transport layer** acts as the primary data highway. It moves sensor data from the perception layer to the processing units and vice versa using technologies such as Wi-Fi, 5G, Bluetooth and NFC. Directly above sits the **processing layer**, commonly referred to as the middleware layer. This critical layer collects incoming data streams and performs extensive

filtering to extract meaningful insights while eliminating redundant or irrelevant information. This directly mitigates the performance bottlenecks typically caused by big data volumes in IoT ecosystems. Finally, the **business layer** sits on top of the architecture and acts as the high-level manager of the entire system. It defines the expected behavior of applications, oversees business and profit models, and enforces user privacy policies by controlling how data is created, stored and modified. However, this centralized logic also makes it a sensitive target, as any vulnerability here could allow attackers to bypass business constraints and abuse the application's core functionality [1, 46].

C) Cloud-Fog-Edge Architecture

Contemporary IoT deployments have increasingly shifted away from strict vertical stacks towards a geographically distributed paradigm known as the **Cloud-Fog-Edge architecture**, driven by real-time industrial constraints. Rather than routing large volumes of raw sensor data directly to a distant, centralized server, this hierarchical topology introduces intermediate fog nodes, such as smart gateways or localized routers, which are positioned closer to the edge devices. While heavy analytical processing and historical data storage are offloaded to the central cloud layer, time-sensitive decisions, localized anomalies and initial data filtering are handled at the network perimeter. As shown in Figure 3, this decentralized infrastructure is specifically designed to minimize network latency, optimize localized bandwidth and prevent the centralized backbone from becoming a single point of failure [1, 46].

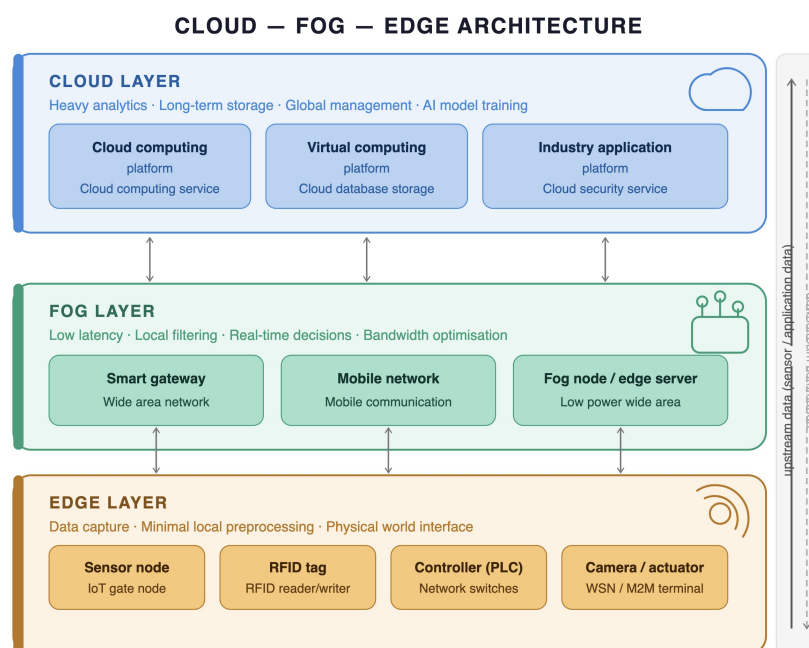


Figure 3: Cloud-Fog-Edge IoT Architecture (adapted from [1])

The **Edge layer** consists of resource-constrained IoT devices — sensors, actuators, and embedded systems — that capture real-world data and perform minimal local preprocessing before forwarding it upstream. The **Fog layer** comprises intermediate nodes, such as smart gateways and local servers, that filter, aggregate, and analyze time-sensitive data in proximity to the source, significantly reducing latency and bandwidth consumption. The **Cloud layer**, finally, provides virtually unlimited storage and computational power for long-term analytics, model training, and global system management. By distributing workloads across these three tiers, this architecture eliminates single points of failure and brings security enforcement closer to where data originates.

3 Threats and attacks in IoT ecosystems

The emergence of new security risks in IoT ecosystems is largely driven by the widespread use of wireless communication protocols. Due to the open nature of these channels, IoT networks are vulnerable to eavesdropping, unauthorized disclosure of information, capture of malicious nodes, and alteration of data. To counter these vulnerabilities, every component within an IoT infrastructure must adhere strictly to the foundational principles of information system security. Specifically, security frameworks must ensure **data confidentiality** to prevent unauthorized access, **data integrity** to guarantee that information is not altered in transit, and **data availability** to ensure that critical services remain accessible to legitimate users [36, 25].

In what follows, we group attacks into three functional domains based on their point of impact: **Edge and Perimeter Attacks** within the Perception Layer (targeting physical hardware and local data), **Communication and Routing Attacks** within the Network Layer (targeting data in transit), and **Platform and Logic Exploits** within the Application Layer (targeting data processing and business applications) [36, 25].

- **Edge and Perimeter Attacks**

Target the perception layer by exploiting the physical vulnerability of resource-constrained sensors through node tampering, cloning, and the deployment of rogue devices that inject falsified data into the network.

- **Communication and Routing Attacks**

Target the network layer where data are in transit. Typical examples include denial of Service (DoS), Hello flood, selective forwarding, Sybil, acknowledgment flooding, Man-in-the-Middle, protocol stack fuzzing, and rogue access points. These attacks exploit the openness of wireless channels to intercept, delay, or corrupt information in transit.

- **Platform and Logic Exploits**

Target the application layer through business logic bypassing, unauthorized data access, and privacy violations that endanger user data and system control across the entire infrastructure [36, 25].

4 Key Challenges in IoT Deployments

4.1 Security and privacy challenges

Securing collected and transmitted data is a critical cybersecurity challenge, yet IoT security is rarely integrated into broader corporate strategies. To resist physical tampering, malware, and network exploits, devices must have baseline defenses. Beyond security, privacy is a major concern, particularly in sensitive sectors like healthcare, defense, and insurance. Today, global data protection laws transform security from an optional technical choice into a strict legal obligation. However, embedding robust encryption into resource-constrained IoT nodes remains costly and complex. To save time and budget, many organizations adopt cheap, inadequate platforms or skip security entirely, forgetting that a single breach can ruin their reputation. To address these deployment risks, industry initiatives like the FIDO Alliance develop standards to automate and accelerate secure device onboarding [19].

4.2 Technical and operational complexity

Although individual IoT sensors perform basic tasks such as measuring data, the overall system is highly complex. A single sensor is merely the first link in a long chain comprising device firmware, hardware, gateways, network protocols and cloud analytics [46]. Consequently, deploying IoT infrastructure involves a steep learning curve. Organizations must develop a clear strategy before purchasing hardware to ensure they anticipate the technical issues that arise when integrating new IoT devices into legacy IT and operational networks.

4.3 Connectivity and energy dependency

Many IoT devices require continuous power and internet connectivity to function reliably. If either of these fails, so does the device and everything connected to it. Given the interconnected nature of IoT systems, entire sub-networks can become non-operational. Companies must understand how failures will affect their hardware in order to anticipate them proactively, since no platform is infallible. Maintenance and incident management processes help to mitigate this problem, as does ensuring that employees know what to do when devices are out of service.

In this regard, edge computing and improvements in battery life are gradually reducing these difficulties [14].

4.4 Protocol Heterogeneity and Integration

As there is currently no consensus on IoT protocols and standards, devices produced by different manufacturers may not be compatible with existing technologies. Each device may require different configurations and hardware connections, which makes efficient deployment difficult. In an industrial environment, the majority of publishers and manufacturers are pushing for the standardization of the OPC UA (Open Platform Communications Unified Architecture) protocol for machine to machine communication [14]. However, physical machine connectivity is only part of the problem. Various network technologies, such as cellular, Wi-Fi, and LoRaWAN, must coexist and share data seamlessly. Organizations must thoroughly analyze this heterogeneous landscape prior to deployment to avoid communication bottlenecks during active operations [46].

4.5 Vulnerability to Device Hijacking

Real-world attacks demonstrate that IoT devices are easy to hijack. Hackers have taken control of smart cameras and thermostats to spy on users, send false emergency alerts, and raise room temperatures to dangerous levels. These incidents rarely stem from flaws in wireless protocols. Instead, they typically arise because users reuse weak passwords across multiple services. When hackers obtain these credentials, they can immediately access the connected IoT devices. Making matters worse, unlike smartphones, standard IoT devices do not generate security alerts, meaning users remain unaware of a breach until significant damage has already occurred [5].

5 Conclusion

In conclusion, the deployment of a secure IoT infrastructure poses significant challenges due to high financial costs, technical complexity, and protocol fragmentation. Furthermore, severe energy dependencies and poor credential management leave many edge devices vulnerable to cyberattacks. As standard IoT endpoints lack built-in security features, they are unable to protect themselves or alert users in the event of a breach. This operational gap demonstrates that traditional security methods are insufficient, creating an urgent need for intelligent, lightweight defense mechanisms capable of protecting resources and detecting anomalies in real time.

CHAPTER 3

Technologies for IoT System Security

1 Introduction

The previous chapter has shown that IoT systems face serious security problems, especially with centralized authentication. These problems include single points of failure, scalability issues, and heavy reliance on a single trusted authority. To address these limitations, we present in this chapter the main technologies that can make IoT ecosystem more secure.

First, we discuss authentication and access control, which are the basic building blocks of any security system. We explain why traditional approaches do not work well for IoT devices with strict resource constraints. Next, we evaluate blockchain technology as a serverless, immutable alternative, focusing on consensus mechanisms and smart contracts tailored for IoT architectures. Third, we introduce artificial intelligence (AI), ranging from classic machine learning to deep learning, and we explain how AI can be combined with blockchain to create a system that can detect threats and respond automatically. Finally, we review existing work to identify research gaps and position our architectural contribution.

This chapter provides the technical background needed to understand the system proposed in the rest of this thesis.

2 Authentication and Access Control Concepts

Authentication and access control are fundamental to securing the IoT, an ecosystem characterized by a rapidly expanding network of resource-constrained and heterogeneous devices. These mechanisms ensure that only authorized entities can interact and access specific resources, which is crucial since IoT nodes often operate autonomously and exchange sensitive data. While authentication verifies the identity of devices and users, access control restricts their permissions based on predefined rights, thereby preserving data confidentiality, integrity, and availability. In this highly dynamic environment, these security pillars must be specifically tailored to overcome the challenges of hardware heterogeneity, strict resource limitations, and massive network scalability [50].

2.1 Traditional Authentication and Access Control Models and Their Limitations for IoT

Identity management is the first line of defense in network security, ensuring that only verified entities can connect and access resources. Traditional frameworks rely on authentication methods such as passwords, certificate-based protocols like X.509, and token-based systems such as JWT and OAuth. They also extend to biometrics, multi-factor authentication, trusted execution environments, and physically unclonable functions (PUFs) for unique device fingerprinting, while access control models such as DAC, MAC, RBAC, and ABAC determine what each authenticated entity is allowed to do [4, 40].

However, deploying these traditional models in the IoT introduces major technical limitations. Standard cryptographic routines and complex access-evaluation algorithms demand substantial memory, computation, and communication resources, creating heavy overhead for resource-constrained, battery-powered IoT devices [41, 15]. In addition, the extreme heterogeneity of the IoT landscape, where devices from different manufacturers use fragmented communication protocols and have widely varying computing capabilities, makes it difficult to deploy a single uniform authentication method [4, 40, 22]. Finally, traditional models scale poorly: when large numbers of dynamic IoT nodes attempt to connect simultaneously, centralized authentication architectures can become congested, causing latency, degraded performance, and reduced reliability [32].

2.2 The Shift Towards Next-Generation Paradigms

The technical limitations and vulnerabilities of centralized infrastructures necessitate a decisive shift toward decentralized, intelligent security paradigms. Relying on a central server to

authenticate large-scale IoT networks creates a critical single point of failure, since a hardware fault or cyberattack on that authority could disrupt the entire system. Furthermore, centralized architectures force all devices to trust one entity, meaning that if that entity is compromised, the wider ecosystem is also compromised. Credential revocation can also be slow and inefficient, leaving the network temporarily exposed [4, 41, 22].

Next-generation paradigms address these structural bottlenecks by decoupling identity management from central authorities and integrating DLT (Distributed Ledger Technologies) and advanced cryptographic methods. Self-Sovereign Identity and Decentralized Identifiers reduce dependence on a single authority by enabling autonomous verification through a shared, tamper-resistant ledger. Meanwhile, zero-knowledge proof-based schemes support secure authentication without exposing sensitive information [50].

However, decentralized identity management alone is not sufficient. Static access control policies cannot detect a device that has already been compromised using valid credentials, so this shift should be paired with continuous behavioral monitoring. Therefore, combining decentralized trust management with lightweight machine learning offers a strong basis for next-generation architectures that can protect resources and detect anomalies in real time [32].

3 Distributed Ledger Technologies and Blockchain in IoT

In this section, we highlight the crucial role of Distributed Ledger Technologies (DLT), particularly blockchain technology, in improving the security and robustness of IoT systems. By shifting away from traditional centralized frameworks, this paradigm offers new mechanisms to enforce integrity, trust, and resilience across large-scale device networks.

3.1 Blockchain and Smart Contract Fundamentals

A blockchain is defined as a decentralized, distributed, and immutable digital ledger that records transactions across a peer-to-peer network [57]. Unlike traditional databases controlled by a single administrative entity, a blockchain inherently distributes exact copies of the ledger to every participating node in the network. This architecture relies on four fundamental properties that make it uniquely suited for applications requiring trust without a central authority [57]. First, **decentralization** removes the reliance on a single point of control. Second, **immutability** ensures that once data is cryptographically recorded, it cannot be altered or deleted. Third, **transparency** allows all authorized participants to audit and verify ledger entries. Finally, **consensus** guarantees that all network nodes agree on the validity of the recorded transactions.

From a structural perspective, a blockchain organizes data into a sequential chain of blocks,

where each block consists of three main components (Figure 4). The *Block Header* stores essential metadata, including a timestamp of its creation, a cryptographic nonce used during the mining validation process, and a Merkle root representing the combined hash of all enclosed transactions. Crucially, the header also contains the cryptographic hash of the previous block, physically chaining the blocks together to prevent retroactive tampering. This header is cryptographically linked to the *Block Body*, which contains the actual list of validated transactions.

In the context of the IoT, the most significant extension of this immutable infrastructure is the smart contract. A smart contract is a self-executing program deployed on the blockchain that automatically enforces predefined rules when specific conditions are satisfied. Rather than relying on a central authority to evaluate and authorize access requests, smart contracts can autonomously manage access control policies. Every transaction or access request processed by a smart contract is transparently verified by network nodes and permanently recorded, eliminating blind trust dependencies and mitigating single points of failure [24].

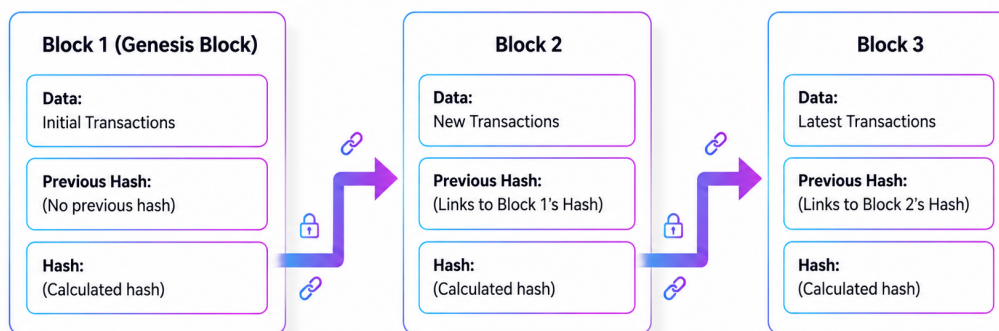


Figure 4: Blockchain Structure

3.2 Consensus Mechanisms in IoT Systems

In a distributed network, nodes must collectively agree on the state of a shared ledger without relying on a central authority. This agreement is achieved through consensus mechanisms designed to solve the Byzantine Generals Problem, ensuring network security even when some nodes fail or act maliciously [12]. When integrating blockchain with the IoT, selecting the appropriate consensus protocol becomes a critical architectural choice, as it directly dictates transaction latency, power consumption, and network scalability.

Traditional consensus algorithms are highly impractical for resource-constrained IoT deployments. For instance, Proof of Work (PoW) requires nodes to solve intensive mathematical

puzzles to validate blocks [39]. This competitive mining process results in exceptionally slow transaction throughput and high energy consumption, which quickly depletes the battery life of small sensor nodes. While Proof of Stake (PoS) improves energy efficiency by selecting validators based on the amount of cryptocurrency they “stake”, it introduces high economic complexity and demands continuous network participation, severely straining the memory and processing limitations of edge devices.

To overcome these constraints, modern IoT architectures leverage lighter, more efficient consensus alternatives:

- **Delegated Proof of Stake (DPoS):** Token holders elect a limited number of dedicated delegates to validate transactions, significantly increasing processing speed and reducing energy overhead.
- **Practical Byzantine Fault Tolerance (PBFT):** Nodes exchange messages in a structured, multi-round voting process to achieve rapid transaction finality. While throughput-efficient, its high communication complexity limits its deployment to small-scale networks.
- **Proof of Authority (PoA):** Block creation is restricted to a predefined set of pre-approved, reputable validator nodes. By entirely eliminating competitive mining and heavy message-voting loops, PoA achieves very fast block times with negligible computational and energy overhead.

Consequently, PoA and DPoS emerge as the most viable consensus mechanisms for large-scale IoT ecosystems. They successfully deliver real-time verification speed, energy conservation, and cost-effective processing, allowing decentralized blockchain networks to scale efficiently within resource-constrained environments [13, 51, 55, 59].

4 Blockchain-Based Authentication

A blockchain-based identity management framework fundamentally restructures how IoT devices register, verify, and terminate their credentials within a network. By shifting the root of trust from a single server to a distributed ledger, the architecture decentralizes identity verification. As illustrated in Figure 5, the operational workflow of this paradigm typically progresses through three distinct sequential phases [45]:

- **Registration:** During the initial deployment, each IoT device generates a cryptographic public-private key pair. The public key is permanently stored on the blockchain ledger alongside a unique device identifier, establishing an immutable digital identity [45].

- **Authentication:** When a device requests access to network resources, it initiates a challenge-response protocol. The device proves its identity by cryptographically signing a challenge message with its private key. A specialized smart contract then autonomously verifies this signature using the corresponding public key retrieved from the blockchain [31, 10].
- **Revocation:** If a device is detected as malicious or compromised, its access must be terminated. The system administrator invokes a specific smart contract function that flags or completely removes the device's public key from the ledger, ensuring instantaneous, network-wide isolation [45].

Consequently, transitioning from traditional centralized architectures to a blockchain-based authentication framework yields significant improvements in structural security, resilience, and operational trust.

First, the decentralized nature of the ledger ensures that authentication is not controlled by a single entity or group. By distributing verification across multiple network nodes, the framework eliminates single points of failure (SPoFs), ensuring that if one part of the system fails, the entire authentication infrastructure remains operational [31, 58]. Second, the cryptographic immutability of the blockchain guarantees that identity data cannot be altered or deleted once recorded, while its inherent transparency provides a fully auditable, open history of all authentication attempts [31, 57]. Finally, the decentralized model scales effectively to accommodate millions of dynamic nodes [16], and the automation provided by smart contracts allows for rapid, automated mitigation when a device is compromised [10].

Despite these clear advantages, integrating blockchain into the IoT landscape introduces critical technical limitations that remain active areas of contemporary research:

- **Latency:** Distributed consensus and ledger lookups can introduce transaction propagation delays, making it naturally slower than immediate, localized centralized authentication [60].
- **Storage constraints:** Blockchains are subject to continuous data inflation. Storing growing transaction histories and massive device registries is highly expensive and easily exceeds the physical memory limits of edge devices [57].
- **Computational overhead:** Executing advanced cryptographic algorithms and signature verifications requires significant processing power, which strains resource-limited nodes [54].
- **Key management:** Because security relies entirely on cryptography, securely storing and managing private keys on lightweight, vulnerable hardware devices remains an open challenge [30].

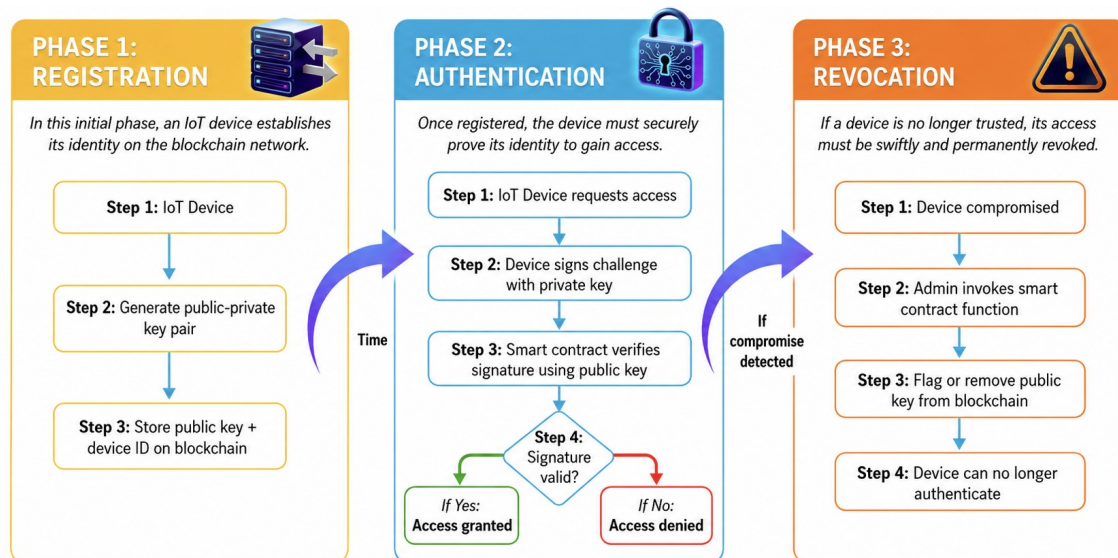


Figure 5: Blockchain-based authentication process

A comparative analysis between blockchain-based authentication and traditional centralized authentication methods is presented in Figure 6

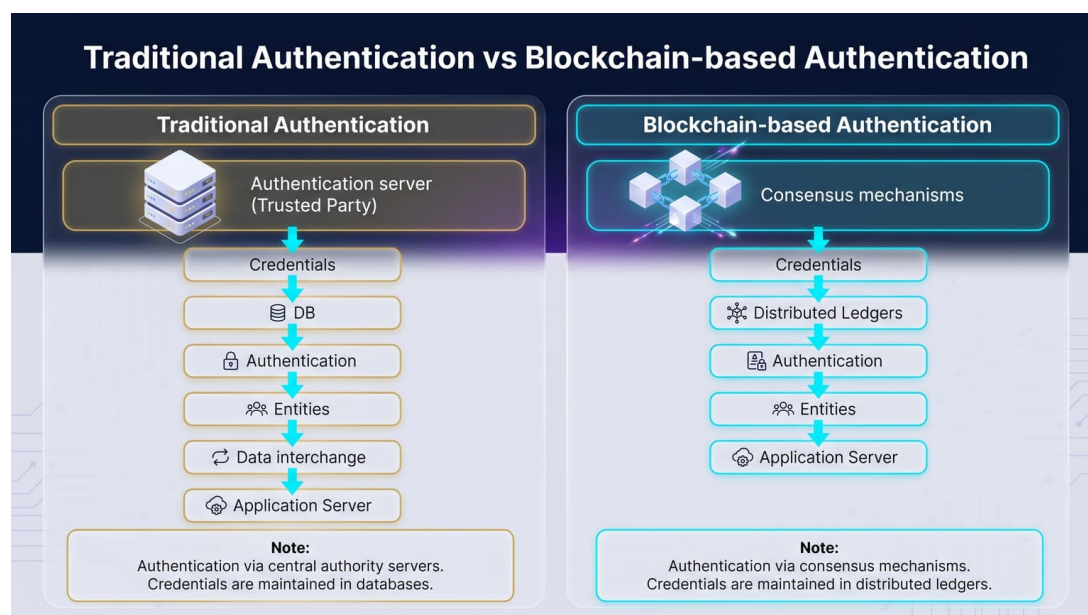


Figure 6: Blockchain-based authentication VS traditional centralized authentication(adapted from [3])

5 Artificial Intelligence for Anomaly Detection in IoT

Although security tools such as blockchain protect the network entrance, they cannot prevent a device from being compromised once it has logged in. To detect these internal threats, modern networks use artificial intelligence (AI) to analyze device behavior. By monitoring network

traffic and device activity in real time, AI can automatically detect unusual behavior and prevent cyberattacks from spreading.

5.1 Taxonomy of AI-Driven Detection Techniques

To systematically evaluate their operational capabilities, AI-driven anomaly detection techniques can be classified into three main categories based on their underlying algorithmic architecture: Machine Learning (ML), Deep Learning (DL), and Hybrid Methods.

A. Machine Learning (ML) Methods

Machine learning methods use simpler, less complex algorithms. These methods are fast, require little memory, and are ideal for small network devices such as routers and home gateways.

- **Isolation Forest:** Anomalies are easier to isolate than normal points. The algorithm builds random trees and measures the isolation depth of each point [43]. Advantage: Lightweight, efficient for high-dimensional data, suitable for resource-constrained IoT environments [26].
- **One-Class SVM:** Restricts normal data space within a boundary. Any point outside is considered an anomaly [26]. Advantage: Efficient for medium-dimensional data [26].
- **Random Forest & XGBoost:** Ensemble methods that combine multiple decision trees for robust classification.
- **Bayesian Networks:** Probabilistic model of cause-effect relationships between different IoT system variables [26]. Advantage: Provides dynamic confidence scores for each detection.

B. Deep Learning (DL) Methods

- **Autoencoders (AE):** A neural network that learns to compress and then reconstruct input data. The reconstruction error helps detect anomalies [26]. Advantage: Unsupervised learning, capable of detecting new types of attacks without labeled data.
- **Convolutional Neural Networks (CNN):** Analyzes spatial relationships in network communications [43]. Advantage: Identifies suspicious network topologies and detects abnormal communication patterns.
- **Long Short-Term Memory (LSTM):** Captures temporal dependencies in IoT device behavior [43]. Advantage: Detects deviations in normal cycles. An LSTM-GRU based IDS achieved 99.01% accuracy on the CIC-DDoS 2019 dataset [43].

- **Proximal Policy Optimization (PPO):** Reinforcement learning to optimize attack response. The agent learns through trial and error [43]. Advantage: The system automatically learns the optimal strategy for each attack type.

C. Hybrid and Advanced Methods

- **Hybrid Models (CNN-LSTM, DAE-ELM):** Combines multiple approaches to improve robustness. CNN-LSTM fuses spatial feature extraction (CNN) with temporal dependency analysis (LSTM) [43]. Advantage: Better robustness and detection accuracy [26].
- **Ensemble Methods (Random Forest, XGBoost):** Combines multiple classifiers to improve robustness [26]. Advantage: Reduces overfitting, better generalization on new data [26].
- **Transfer Learning:** Reuses knowledge acquired on one attack type to detect another [43]. Advantage: Fast learning (fine-tuning) instead of full retraining, crucial for resource-constrained IoT environments.

5.2 How to Link AI with Blockchain

The intersection of blockchain and AI overcomes the fundamental limitations of both technologies. Blockchain provides decentralized trust and immutable data, and AI offers the predictive intelligence necessary to detect adaptive network threats. Together, they enable a robust, transparent, and self-healing anomaly detection ecosystem, which can be summarized as follows:

- **Decentralized Model Aggregation:** Blockchain coordinates local model aggregation without a central server, eliminating the single point of failure [43].
- **Blockchain for AI Model Integrity:** Detection parameters and scores can be recorded on the blockchain, ensuring immutable traceability and proof of integrity for AI decisions [43].
- **Smart Contracts for Automated Response:** When an anomaly is detected, a smart contract can automatically trigger countermeasures: isolating compromised devices, revoking their keys, or sending alerts [43].
- **Federated Learning on Blockchain:** Blockchain serves as a ledger for federated model updates, ensuring contribution authenticity and preventing poisoning attacks [43].
- **Incentive Mechanisms for Data Sharing:** Tokens can reward devices that participate in training or report valid alerts, encouraging secure collaboration [43].

In summary, blockchain provides trust, decentralization, and immutability, while AI provides intelligence and adaptability. Their combination enables a robust, transparent, and automated anomaly detection system [43, 26].

6 Analysis of Existing Work

This section reviews ten representative contributions from the recent literature on IoT security, organized around three thematic pillars: lightweight cryptography and authentication, blockchain-based decentralized architectures, and machine learning-driven intrusion detection. Each work is presented as a short narrative paragraph combining the authors, year, and a description of its contribution, rather than as a structured title/category block.

6.1 Lightweight Cryptography and Authentication Mechanisms

Dhanda et al.(2020) [14], focus on lightweight cryptography and authentication by providing a comprehensive comparative study of cryptographic algorithms tailored for resource-constrained IoT devices. The authors evaluate the performance of prominent hardware- and software-optimized ciphers, including PRESENT and SPECK, by analyzing their energy consumption, memory footprint, and resistance to cryptanalysis. This work establishes clear selection criteria, helping developers choose the right cryptographic solutions based on the specific hardware limits and security needs of their IoT environment.

Panda and Chattopadhyay, (2020) [42], address the problem of device-to-device trust by proposing a secure mutual authentication mechanism that ensures reliable communication and cryptographic key exchange between IoT devices. Their protocol is specifically designed to secure device-to-device communication without relying on pre-shared secrets or a centralised trust anchor, making it well suited to dynamic, infrastructure-light IoT deployments.

Khan et al.(2020) [30], target the particularly sensitive context of medical IoT by improving elliptic curve cryptography (ECC) implementations for healthcare applications. Their framework jointly provides authentication and encryption tailored to environments where patient data confidentiality is paramount, demonstrating that lightweight cryptography can be adapted to domain-specific privacy requirements.

Thakor et al.(2021) [54], provide a broader comparative perspective by reviewing and benchmarking DES, AES, RSA, and ECC across resource-constrained IoT devices. Their analysis quantifies the trade-offs between security strength and performance metrics, offering practical guidance for selecting cryptographic primitives in constrained device deployments.

6.2 Blockchain-Based Decentralized Authentication and Privacy

Rashid and Pajoo, (2019) [45], develop a blockchain-based authentication and authorization framework that relies on smart contracts to manage IoT device identities. By enforcing self-executing access control policies directly on a distributed ledger, their approach eliminates dependence on a centralised authority, addressing the single-point-of-failure weakness inherent to traditional IoT authentication architectures.

Fan et al.(2023) [16], combine blockchain technology with federated learning to protect data privacy during collaborative model training in IoT environments. Their contribution is particularly relevant to large-scale deployments, where sensitive data must remain on local devices while still contributing to a shared, privacy-preserving learning process.

Zhang et al.(2022) [58], implement a secure decentralized IoT service platform built on consortium blockchain technology. Their architecture targets enterprise deployments where participant identities are known in advance but organisational trust boundaries must nonetheless be enforced, illustrating how permissioned blockchains can balance openness and control.

6.3 Machine Learning for Intrusion Detection and Vulnerability Analysis

Feng et al. (2022) [18], survey firmware analysis techniques for detecting security vulnerabilities in IoT device firmware. By combining static and dynamic examination methodologies, their work strengthens device-level protection at a stage that is often overlooked compared with network-level defences.

Bast and Yeh, (2024) [7], explore the implementation of zero trust architectures for IoT environments, arguing that traditional perimeter-based security models are inadequate for highly distributed, dynamically changing deployments. Their work reinforces the broader shift toward continuous verification rather than one-time perimeter authentication.

Seyed et al. (2024) [47], develop an automatic intrusion detection mechanism that employs CNN, LSTM, and MLP architectures in combination. Evaluated on benchmark datasets including NSL-KDD and CIC-IDS2017, their system achieves detection rates between 98.5% and 99.5%, demonstrating the strong potential of hybrid deep learning architectures for IoT intrusion detection.

Table 1 summarizes the technical approaches, contexts, and limitations of these baseline works, mapping out the current state of the art.

Table 1: Summary of Existing Works on IoT Security Technologies (2019–2024)

Ref	Year	Technique	Algorithm	Dataset	Description	Limitation
[14]	2020	Lightweight Cryptography	AES, PRESENT, SPECK	—	Comparative survey of lightweight ciphers for constrained IoT devices	No empirical validation on real IoT hardware
[42]	2020	Authentication Protocol	Mutual Authentication Protocol	—	Secure mutual authentication for IoT environments	Lacks large-scale deployment testing
[30]	2020	Medical IoT Security	ECC	Medical sensor data	Improved ECC for medical IoT sensor data	Limited to healthcare-specific scenarios
[45]	2019	Blockchain Authentication	Smart Contracts	—	Blockchain-based authentication framework using smart contracts	No AI-based anomaly detection layer
[16]	2023	Blockchain + Federated Learning	Federated Learning	IoT data	Lightweight privacy-preserving blockchain with federated learning	High communication overhead at scale
[18]	2022	Vulnerability Detection	Firmware Analysis	IoT firmware samples	Survey of vulnerability detection in IoT firmware	Does not address runtime detection
[7]	2024	Zero Trust Authentication	Emerging Authentication Protocols	—	Zero trust architecture for IoT systems	No concrete implementation provided
[54]	2021	Lightweight Cryptography	DES, AES, RSA, ECC	—	Review of lightweight cryptographic algorithms	Comparative only, no new protocol proposed
[58]	2022	Blockchain Security	Consortium Blockchain	IoT service data	Secure decentralized IoT service platform	Requires known, pre-vetted participants
[47]	2024	Intrusion Detection	CNN, LSTM, MLP	NSL-KDD, CIC-IDS2017	Automatic IoT intrusion detection using machine learning	Centralized training limits scalability

6.4 Synthesis and Positioning

The reviewed literature reveals three important trends. First, lightweight cryptography remains an active research area, with continuous optimisation aimed at reducing device footprint while preserving security guarantees [14, 54]. Second, blockchain integration has progressively shifted from purely theoretical proposals to practical implementations that explicitly address IoT-specific constraints such as resource limitation and trust distribution [45, 58]. Third, machine learning approaches achieve impressive detection accuracies [47], but they often rely on centralised training pipelines that may not scale efficiently to massively distributed IoT deployments.

Our work positions itself at the intersection of these three streams. Unlike [14, 54], which focus exclusively on cryptographic primitives, we integrate lightweight ECDSA with zero-knowledge proofs to enable privacy-preserving authentication. While [45, 58] employ blockchain

for decentralised identity management, our architecture further adds AI-powered anomaly detection coupled with automated smart contract revocation, a feedback loop that is absent from prior work. Finally, where [47] achieves high detection rates through centralised training, our ensemble BiLSTM operates within a decentralised context in which model updates could be federated across the blockchain network, aligning with the direction taken by [16] but extending it toward real-time operational deployment.

7 Conclusion

In conclusion, although the current state of the art offers robust, isolated solutions for cryptographic verification, decentralized consensus and deep learning classification, a unified framework that balances all three of these properties within the context of resource-constrained IoT environments is still lacking.

To address this issue, Chapter 4 will present our suggested security architecture. The proposed framework leverages a lightweight ECDSA protocol combined with Zero-Knowledge Proofs (ZKPs) for privacy-preserving authentication, a Proof-of-Authority blockchain for decentralized identity management, and a BiLSTM autoencoder for real-time anomaly detection, with smart contracts automating response and revocation.

CHAPTER 4

Proposed Architecture, Implementation and Evaluation

1 Introduction

This chapter presents the practical realization of our decentralized IoT authentication system. Having established the theoretical foundations and reviewed the state of the art in the previous chapters, this chapter focuses on the concrete implementation of the proposed solution. It covers the global system architecture, the justification of the core technology choices, the layer-by-layer technical implementation, the system integration with attack simulations, and a comprehensive performance evaluation.

The chapter is organized as follows. Section 2 presents the global four-layer system architecture. Section 3 details the core technology choices and their justifications. Section 4 describes the technical implementation of each layer. Section 5 covers system integration, security scenarios, and the monitoring dashboard. Section 6 provides a comprehensive performance evaluation and discussion. Section 7 concludes the chapter.

2 Proposed System Architecture

In order to balance the use of intensive central detection engines with the resource limitations of edge nodes, the proposed framework employs a decoupled, four-layer architecture. This design maintains an immutable, reliable audit trail of network events and utilizes advanced AI-based threat detection, while keeping local IoT devices lightweight.

Figure 7 illustrates the full structure of this four-layer architecture.

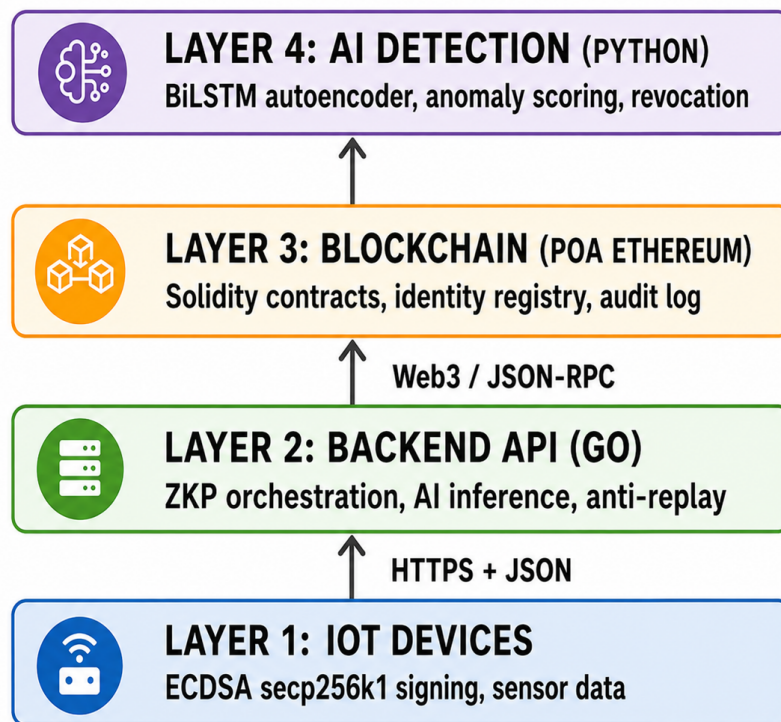


Figure 7: Four-Layer IoT System Architecture

The architecture is organized into four functionally independent layers, each with clearly defined responsibilities:

2.1 IoT Devices Layer

Resource-constrained sensors perform only lightweight ECDSA signature generation and ZKP proof submission locally. All heavy computation is delegated to the backend.

2.2 Backend API Layer

Receives signed messages, orchestrates ZKP verification, enforces anti-replay checks (timestamp, sequence, nonce), and forwards authenticated events to the blockchain.

2.3 Blockchain Layer

A private Proof-of-Authority network with 4 validators and 4 full nodes stores device identities and authentication records immutably via a single smart contract (IoTDeviceManager).

2.4 AI Detection Layer

A BiLSTM Autoencoder service analyzes 14 network traffic features asynchronously after each authentication, computing an anomaly score and triggering automatic device revocation when the score exceeds the learned threshold.

2.5 Inter-Layer Communication

Each layer communicates with the adjacent ones through well-defined interfaces. IoT devices submit signed payloads over HTTPS to the Backend API. The backend records authenticated events on-chain via Ethereum JSON-RPC (Layer 2 $\rightarrow$ Layer 3), and asynchronously forwards network traffic features to the AI detection Layer (Layer 2 $\rightarrow$ Layer 4). Upon anomaly detection, the AI service triggers an on-chain revocation call back through the backend (Layer 4 $\rightarrow$ Layer 2).

3 Core Technology Choices

This section justifies the selection of the three core technologies integrated into the proposed framework: ECDSA for device authentication, Zero-Knowledge Proofs for identity confidentiality, and a BiLSTM Autoencoder for anomaly detection.

3.1 ECDSA for Lightweight IoT Authentication

ECDSA has emerged as the predominant digital signature standard for constrained environments, offering an optimal balance between security strength and computational efficiency [28, 29]. Its primary implementation risk lies in improper nonce generation, which can lead to private key exposure [8]. Recent studies confirm its suitability for IoT, as summarized in Table 2.

Table 2: Recent Studies Validating ECDSA for IoT Authentication

Study	Key Metric	Findings
Thakor et al. (2021) [54]	Efficiency vs. RSA	ECDSA provides approximately $4.5\times$ efficiency advantage over RSA at equivalent security level, with substantially shorter key sizes
Benabderrezak et al. (2025) [8]	Throughput on Constrained Hardware	Practical feasibility demonstrated with 20,366 signatures per minute on constrained hardware
Shaaban, Mohamed Ali, et al. (2024) [48]	Blockchain Compatibility	ECDSA on secp256k1 enables native Ethereum blockchain compatibility with minimal on-device overhead

We use ECDSA on the secp256k1 curve, which is standardized by Ethereum [56], to ensure native blockchain compatibility. IoT devices only perform lightweight signature generation locally, while the computationally intensive verification process is delegated to the backend. This preserves limited device resources without compromising security.

3.2 Zero-Knowledge Proofs for Identity Confidentiality

ZKP protocols enable a prover to demonstrate knowledge of a secret without disclosing it [21, 17], a property particularly relevant to IoT authentication where devices must prove identity without exposing credentials over hostile channels. Recent research validates their applicability, as summarized in Table 3.

Table 3: Recent Studies Validating ZKP for IoT Authentication

Study	Key Metric	Findings
Soni & Singh (2025) [52]	Verification & Cost	ZKP-based smart contract authentication achieves fast verification with significant operational cost reduction in IoT smart home environments
Boo et al. (2022) [9]	IoT Edge Performance	LiteZKP reduces energy consumption and latency of ZKP-based blockchains on IoT and edge platforms by more than 55%
Ramezan & Meamari (2024) [44]	Lightweight ZKP on IoT	zk-IoT demonstrates that modern proof systems (Bulletproofs+, Halo2) can effectively run on low-power IoT devices with seconds-level verification

Our framework integrates the Groth16 proving system [23] combined with the MiMC hash function [2], yielding 991 R1CS constraints and 192-byte proofs. ZKP proof generation is performed exclusively on the backend, ensuring IoT devices remain lightweight while benefiting from zero-knowledge privacy guarantees.

3.3 BiLSTM Autoencoder for Anomaly Detection

The BiLSTM Autoencoder detects anomalous IoT traffic through reconstruction error analysis: higher reconstruction error indicates suspicious behavior [34]. This architecture was selected over alternatives such as Isolation Forest for three main reasons: its unsupervised learning capability (training exclusively on normal data, without requiring labeled attack samples), its bidirectional temporal context capture which is well-suited to sequential network flows, and its zero-day attack detection potential via generalized reconstruction error.

As shown in Table 17 (Section 6.4), the BiLSTM Autoencoder achieves a 22.55 percentage point absolute improvement in F1-score over an Isolation Forest baseline trained under identical conditions, confirming the validity of this architectural choice for real-time IoT anomaly detection.

4 System Implementation and Security Mechanisms

4.1 Cryptographic Layer (Layers 1 & 2)

4.1.1. ECDSA Signature Generation (secp256k1 Curve)

ECDSA operates on an elliptic curve defined by $y^2 = x^3 + 7$ over a finite field $\mathbb{F}_p$, where $p = 2^{256} - 2^{32} - 977$ [56]. The security relies on the computational hardness of the Elliptic Curve Discrete Logarithm Problem (ECDLP) [28, 29].

The hybrid ECDSA–ZKP architecture creates a dual-layer security model addressing complementary threats. ECDSA provides lightweight on-device authentication and non-repudiation [28], while ZKP ensures identity confidentiality and anti-replay protection by binding each signature to a unique session without revealing private components [23]. Table 4 compares the security guarantees of each approach.

Table 4: Security Comparison: ECDSA Alone versus ECDSA + ZKP

Attack Vector	ECDSA Alone	ECDSA + ZKP
Identity Spoofing	✓	✓
Replay Attack	✗	✓
Man-in-the-Middle	~	✓
Private Key Compromise	✗	✓
Passive Eavesdropping	✗	✓

✓ = Protected ✗ = Vulnerable ~ = Partially Protected

The 65-byte signature structure is detailed in Table 5. The complete signature generation implementation is provided in Annex 1, and the verification procedure is detailed in Annex 2.

Table 5: Structure of a 65-byte ECDSA Signature (secp256k1)

r (32 bytes)	s (32 bytes)	v (1 byte)
$R = k \cdot G$ (X-coordinate)	Proof of private key knowledge	Recovery ID
Bytes 0–31	Bytes 32–63	Byte 64

4.1.2. ZKP Circuit Design with Groth16 (BN254, MiMC)

The ZKP circuit [23] defines one private input (*Secret*, derived via HKDF-SHA256 from the master secret and the device address) and three public inputs verified by the backend: the device Ethereum *Address*, an anti-replay *Challenge* computed as $\text{Keccak256}(\text{timestamp} \parallel \text{sequence} \parallel \text{nonce})$, and the *ExpectedHash*. The circuit proves that $\text{MiMC}(\text{Secret}, \text{Address}, \text{Challenge}) = \text{ExpectedHash}$ without revealing the secret itself.

The MiMC hash function [2] was selected for its minimal multiplicative complexity, which directly translates into a low R1CS constraint count (991 constraints) and therefore fast proving times. The BN254 (alt_bn128) curve supports efficient pairing operations required by Groth16 [23] and is natively supported by the Ethereum EVM, enabling future on-chain verification without additional libraries.

The 192-byte proof structure on BN254 is described in Table 6. The complete ZKP circuit implementation is provided in Annex 3, and the generation and verification procedures are detailed in Annex 4.

Table 6: Structure of a 192-byte Groth16 Proof on BN254 (uncompressed)

Component	Size	Curve Point	Description
π_A	64 bytes	$\mathbb{G}_1$	First proof element
π_B	64 bytes	$\mathbb{G}_2$	Second proof element
π_C	64 bytes	$\mathbb{G}_1$	Third proof element
Total	192 bytes	$\pi = (\pi_A, \pi_B, \pi_C)$	128 bytes compressed

4.2 Blockchain and Smart Contracts (Layer 3)

4.2.1. PoA Private Ethereum Setup

The blockchain layer relies on a private Ethereum Proof-of-Authority network [13] comprising 4 validator nodes responsible for block creation (approximately 5 seconds per block) and 4 full nodes handling chain synchronization and transaction verification. All nodes run the Geth client and are interconnected within an isolated private network, ensuring that no transactions or state are exposed to the public Ethereum mainnet.

Proof-of-Authority is chosen for its deterministic block finality, negligible energy consumption, and suitability for permissioned IoT deployments where validators are known and trusted entities [13].

4.2.2. Smart Contracts (Identity Registry and Audit Log)

A single smart contract, `IoTDeviceManager` (Solidity, compiled with Hardhat), consolidates all blockchain interactions through three core functions:

- `registerDevice`: stores the device address, public key, metadata, and entity type (IoT device, validator, or fullnode) on-chain, forming the immutable identity registry.
- `recordAuthentication`: logs every authentication event with a success flag and the 32-byte ZKP proof hash, creating a tamper-proof audit log.
- `activateDevice` / `deactivateDevice`: manages the device lifecycle; deactivation is triggered automatically by the AI detection service upon anomaly detection, enforcing a closed-loop revocation mechanism.

The complete device registration and blockchain recording implementations are provided in Annex 6 and Annex 5.

4.3 Deep Learning Engine Implementation (Layer 4)

4.3.1. Feature Engineering (14-Feature Space Construction)

For training and evaluation, we employed three benchmark datasets: **IoT-23** [20] (benign and botnet traffic), **CSE-CIC-IDS2018** [49] (DDoS, botnet, brute force attacks), and **Bot-IoT** [38] (botnet scenarios). A balanced fusion strategy with a 1:2.5 normal-to-anomaly ratio yielded a final dataset of 187,183 samples (53,481 normal, 133,702 anomalies).

We engineered 14 features from raw network flows, detailed in Table 7. Features are grouped into four categories: base flow statistics, derived statistical ratios, log-space transforms, and temporal behavioral indicators.

Table 7: Feature Set Description (14 Features)

Feature	Type	Description
duration	Base	Flow duration (seconds)
bytes_sent	Base	Bytes sent from source
bytes_received	Base	Bytes received
packet_count	Base	Total number of packets
bytes_per_sec	Statistical	Transmission rate
pkts_per_sec	Statistical	Packet rate
bytes_per_pkt	Statistical	Average packet size
bytes_ratio	Statistical	Upload/Download ratio
bytes_sent_ratio	Statistical	Upload proportion
bytes_total	Statistical	Total bytes transferred
log_bytes_total	Transform	Log-transformed total bytes
pkt_size_variance	Statistical	Packet size variability
high_volume_flag	Temporal	Flag for high traffic volume (95th percentile)
burst_indicator	Temporal	Short burst detection (duration < 1 s)

4.3.2. BiLSTM Autoencoder Network Architecture

The BiLSTM Autoencoder detects anomalies through reconstruction error analysis: normal traffic is accurately reconstructed (low MSE), while anomalous patterns produce elevated reconstruction error exceeding the learned threshold [34].

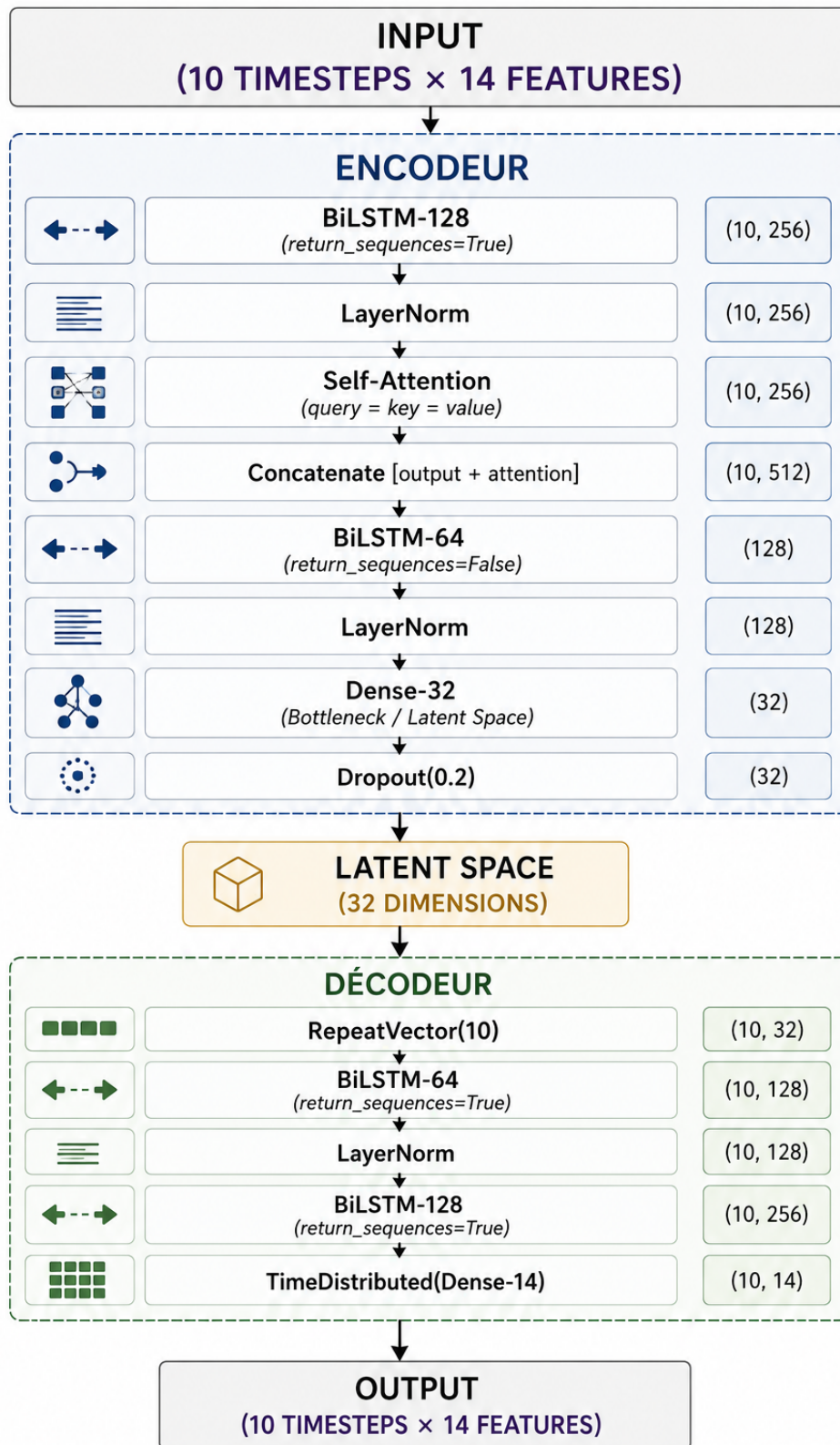


Figure 8: Architecture of the BiLSTM Autoencoder Model

The model consists of three components:

1. **Encoder:** Two bidirectional LSTM layers (128 and 64 units) with a self-attention mechanism compress the input sequence (14 features over 10 time steps) into a latent representation.
2. **Bottleneck:** A dense layer (32 neurons, dropout 0.2) with Layer Normalization creates the most compressed representation, forcing the model to learn the essential patterns of normal traffic.
3. **Decoder:** Two bidirectional LSTM layers (64 and 128 units) reconstruct the original sequence. The Mean Squared Error (MSE) between input and reconstruction serves as the anomaly score.

The complete model contains **763,438 trainable parameters**. It was implemented in TensorFlow 2.19.0 and trained on a Kaggle environment (Tesla P100 GPU). Key hyperparameters include: early stopping (patience 20, best weights restored), ReduceLROnPlateau scheduling (factor 0.5, patience 10, $\text{min_lr} = 10^{-7}$), gradient clipping ($\text{clipnorm} = 1.0$), L2 regularization ($\lambda = 0.001$), dropout 0.35, and batch size 64. Training converged after 127 epochs (early stopping triggered), with a final validation loss of 0.1595. The optimal anomaly threshold was calibrated on the validation set using the $\mu + k \cdot \sigma$ method with $k = 0.5$, yielding a threshold of **0.3425**.

The complete implementation is available at: <https://www.kaggle.com/code/benadl/iot-anomaly-detection>.

5 System Integration, Security Scenarios and Interface

5.1 Automated Smart Contract Revocation Mechanism

The framework implements a closed-loop, automated revocation pipeline that links the AI detection layer directly to the blockchain identity registry. Upon receiving network traffic features from the backend, the BiLSTM inference service computes the reconstruction MSE in real time. If the anomaly score exceeds the calibrated threshold of 0.3425, the service immediately notifies the Go backend, which in turn calls `deactivateDevice` on the `IoTDeviceManager` smart contract. From this point, any subsequent authentication attempt by the flagged device is rejected at step (3) of the 8-step verification chain (on-chain activity check), without requiring manual intervention. This automated revocation mechanism ensures that compromised devices are isolated within seconds of anomaly detection, well before an operator could manually review the event.

5.2 Simulated Real-Time Attack Scenarios

To validate the end-to-end authentication pipeline and the AI detection system, we simulated three distinct attack scenarios on registered IoT devices using a dedicated anomaly simulator.

End-to-End Authentication Flow: The complete authentication pipeline combines all four layers in a coherent two-phase flow:

1. **Phase 1 - ZKP Generation:** The IoT device sends its ECDSA signature and message hash to `POST /api/zkp/generate-secure`. The backend verifies ECDSA, derives the device secret via HKDF-SHA256 (`GenerateSecureSecretFromMaster`), generates and stores the anti-replay challenge, and runs the Groth16 prover. It returns the 192-byte proof and the expected hash to the device.
2. **Phase 2 - Authenticated Message Submission:** The device sends the full payload (message + ECDSA signature + ZKP proof + expected hash + nonce + timestamp + sequence + 14 ML features) to `POST /api/node/message-secure`. The backend applies an 8-step verification chain: (1) timestamp validation (< 300 s), (2) sequence check (strictly monotonic), (3) device registration and activity check on-chain, (4) message hash recomputation, (5) ECDSA verification, (6) challenge reconstruction and atomic consume, (7) ZKP Groth16 verification, (8) nonce marking (only after steps 5 and 7 pass). On success, the backend asynchronously calls the BiLSTM ML service (`go sendToMLService`), records the event on-chain via `AuthRecordWithMeta`, and logs real-time timings for the dashboard.

Three-Profile Attack Simulation: The simulator registers three dedicated attack devices, each assigned a distinct anomaly profile with crafted ML feature values designed to push the BiLSTM autoencoder's reconstruction error well beyond the detection threshold ($MSE > 0.3425$). The three profiles are summarized in Table 8.

Table 8: Attack Profiles and Targeted BiLSTM Severity Levels

Device ID	Profile	Feature Signature		Target
ANOMALY_001	DDoS / Flood	50 000–80 000 burst_indicator = high_volume_flag = 1, 0.5–1 s	packets, 95–100, duration	CRITICAL
ANOMALY_002	Data Exfiltration	bytes_sent bytes_sent_ratio bytes_received pkt_size_variance > 250	500 MB–1 GB, ≈ 1.0, < 5 KB,	HIGH
ANOMALY_003	Port Scan / Recon	8 000–12 000 micro-packets (40–60 B each), pkt_size_variance > 1800, high_volume_flag = 0		MEDIUM

Table 9 details the key ML feature contrasts between normal device traffic and the most severe attack profile (DDoS / Flood).

Table 9: DDoS Attack Traffic Characteristics vs. Normal Ranges

Feature	Normal Range	Attack Value	Ratio
bytes_sent	100–500 B	3.2–6.4 MB	×10 000
packet_count	2–10	50 000–80 000	×8 000
burst_indicator	0–2	95–100	×50
high_volume_flag	0	1	ON
pkts_per_sec	1–5	60 000–130 000	×25 000
Duration	2–5 s	0.5–1 s	÷5

Detection Results: Each attack device passes the full cryptographic pipeline (ECDSA + ZKP Groth16) on every request, since the goal is to test ML-layer detection of semantically anomalous traffic from otherwise authenticated devices. The simulator introduces a ±20% jitter on the 5-second send interval to reproduce realistic network variability.

After accumulating the minimum required buffer (10 messages per the seq_len parameter), the BiLSTM autoencoder produced elevated reconstruction errors on all three profiles. The

dashboard AI Detection panel captured 15 anomalies across 3 monitored devices within the first simulation run, with all 15 classified as **CRITICAL** by the severity scoring function (score > 90, MSE $\gg$ 0.3425), as shown in Figure 9. Automatic device revocation was triggered via POST /api/device/deactivate for each flagged device.

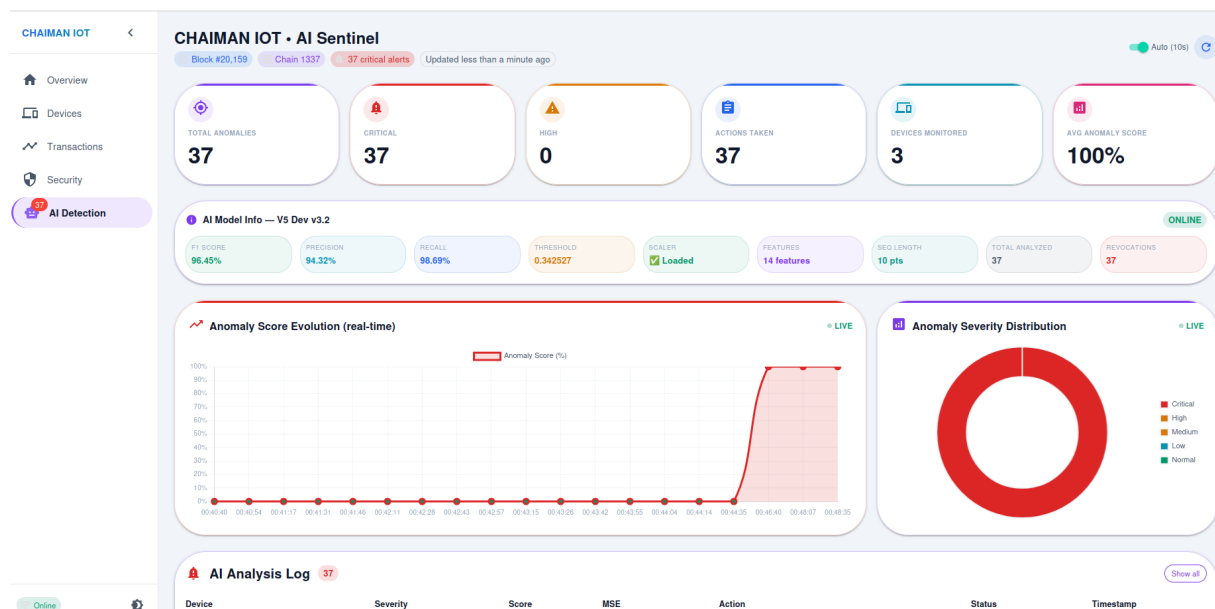


Figure 9: CHAIMAN IoT dashboard — AI Detection panel during the three-profile attack simulation.

Legitimate devices running `iot_simulator.go` maintained a 100% authentication success rate throughout the simulation, confirming that the security layers introduce no false positives on normal traffic.

Table 10: Integrated Security Mechanisms

Mechanism	Description
AI Anomaly Detection	BiLSTM Autoencoder (763,438 params, F1 = 94.40%, Precision = 95.28%, Recall = 93.53%) evaluates 14 network features per message. MSE > 0.3425 triggers automatic device revocation via POST /api/device/deactivate with HMAC-SHA256 authentication.
Three-Profile Attack Coverage	DDoS/Flood (CRITICAL), Data Exfiltration (HIGH), and Port Scan (MEDIUM) profiles each pass full ECDSA+ZKP authentication while injecting semantically anomalous ML features, validating that the detection layer operates independently of the cryptographic layer.
Anti-Replay Protection	Three independent layers: timestamp (< 5 min), strictly increasing sequence numbers, and unique nonces per device. Nonces are marked only after both ECDSA and ZKP verification succeed, preventing nonce-exhaustion DoS attacks.
Rate Limiting (IP + Address)	Three-tier rate limiting : a global per-IP limiter at the HTTP middleware level (10 req/s, burst 20, in production; 1000 req/s, burst 2000, in benchmark mode), and a secondary per-IP/per-device-address limiter at the handler level (50 req/s, burst 100, in production; 1000 req/s, burst 2000, in benchmark mode). This layered approach absorbs traffic spikes before they reach the cryptographic verification logic.
Smart Contract Storage	IoTDeviceManager stores device registrations with entity type separation (IoT device, validator, fullnode).

5.3 Monitoring Dashboard Interface

The React frontend provides five monitoring interfaces, offering real-time visibility across all system layers.

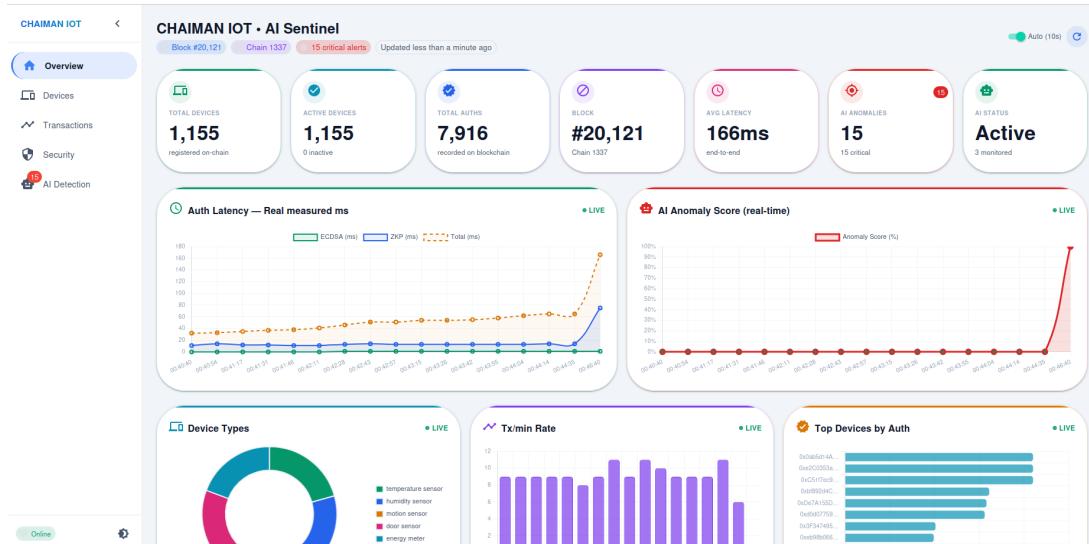


Figure 10: Dashboard Overview

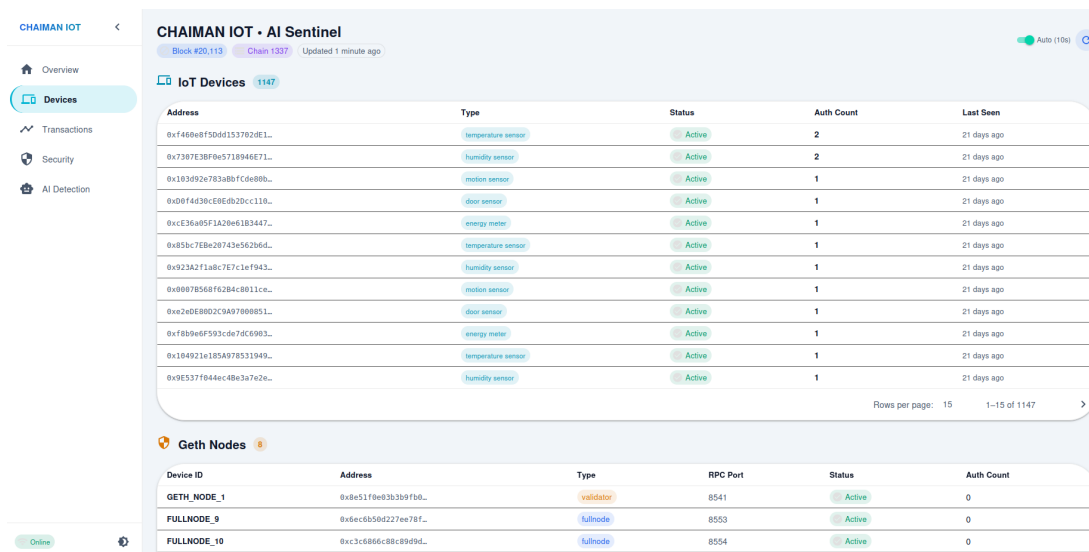


Figure 11: Device Management

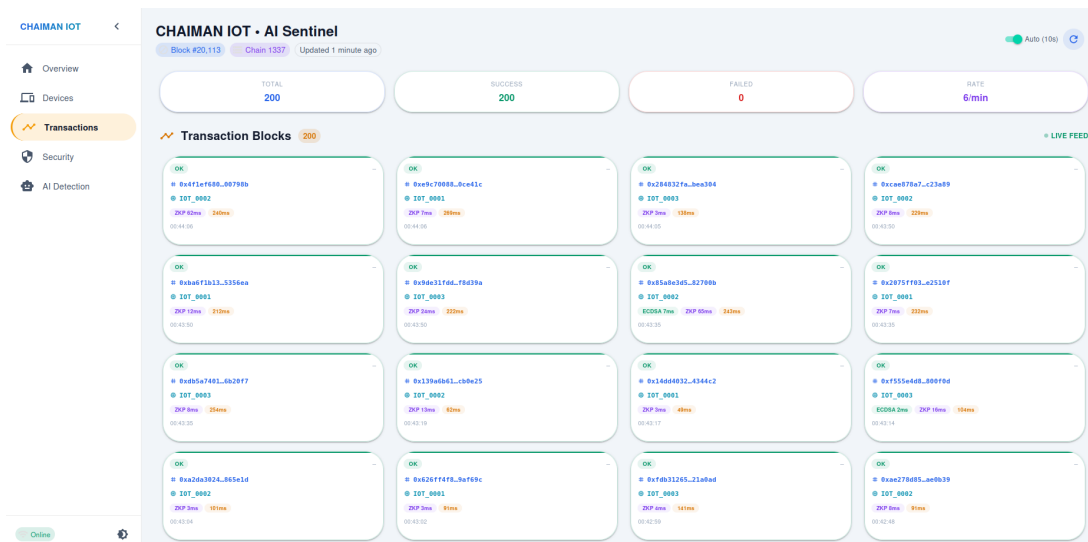


Figure 12: Transactions Dashboard

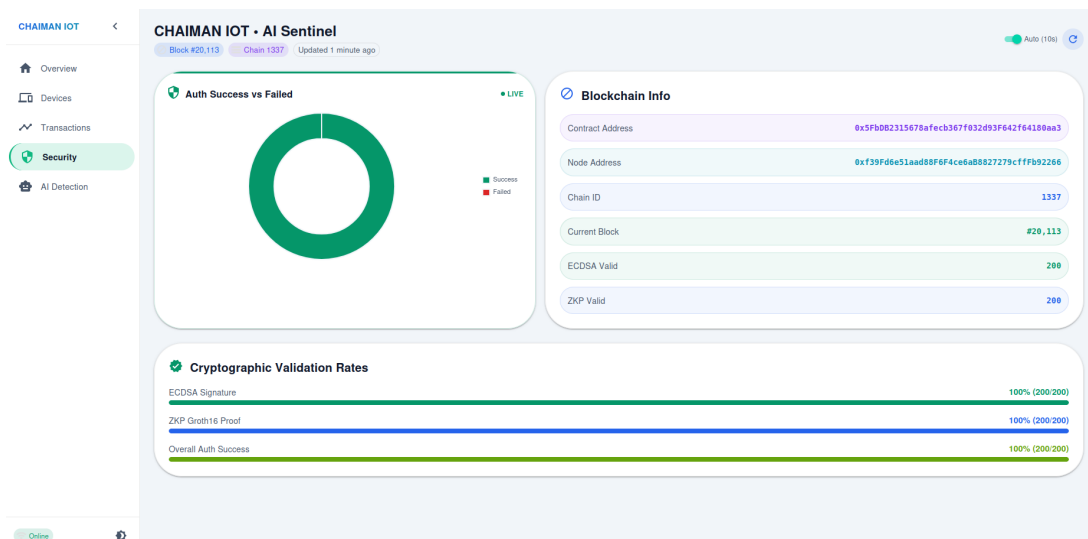


Figure 13: Security Dashboard

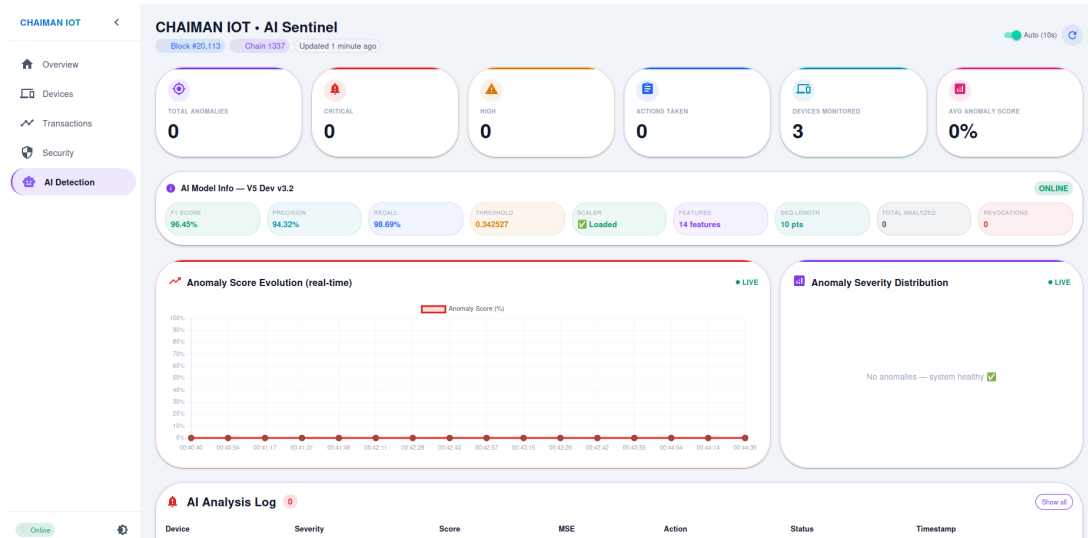


Figure 14: AI Detection Interface

6 Performance Evaluation and Discussion

6.1 Experimental Environment and Deployment Setup

Experiments ran on a private PoA blockchain [13] with 4 validators and 4 full nodes. IoT devices were deployed in five incremental phases (20, 50, 100, 150, and 200 devices) across five sensor types, transmitting every 10–30 seconds. The backend performed ECDSA verification, ZKP generation (Groth16/BN254, 991 R1CS constraints), and blockchain submission. All tests were conducted after optimizing rate-limiting, state synchronization, and request spreading to eliminate test harness artifacts.

6.2 Authentication, ZKP Circuit and Throughput Metrics

6.2.1. Registration and Verification Latency

HTTP latency (T1) dropped from 173.5 ms at P1 to 69.0 ms at P5 as the backend warmed up under sustained load. Blockchain confirmation time (T2) remained stable at approximately 3.5 seconds throughout, consistent with the 5-second block interval and 2-block confirmation depth [13]. Total registration latency stayed between 3.6 and 3.8 seconds regardless of device count.

Before presenting the detailed measurements, it is useful to clarify what each latency component represents. **T1 (HTTP latency)** corresponds to the synchronous request–response round-trip between the calling client (IoT device or dashboard) and the Go backend’s REST API over HTTP(S): it captures network transit time, TLS overhead (when a session is not reused), JSON

payload (de)serialization, in-handler ECDSA/ZKP signature verification, and the time the backend takes to dispatch the transaction to the Ethereum node — but it stops as soon as the backend returns its HTTP response to the client, *before* the submitted transaction is actually mined. **T2 (blockchain confirmation time)**, in contrast, measures the additional time required for that transaction to be included in a block and to reach the required confirmation depth on the PoA chain. Distinguishing T1 from T2 makes it possible to separate API/network-layer overhead — which can be optimized through caching, connection reuse, or backend horizontal scaling — from the latency floor imposed by block production itself, which is bounded by the chain’s consensus parameters [13].

Table 11: Registration Performance by Phase

Phase	Devices	T1 HTTP (ms)	T2 Confirm (ms)	Total (ms)	Gas (units)	Success
P1	20	173.5±128.3	3,486.5±866.8	3,660	332,527	98.0%
P2	50	159.4±64.4	3,533.1±364.6	3,693	331,515	98.2%
P3	100	105.3±63.5	3,592.3±293.0	3,698	331,514	98.6%
P4	150	131.1±85.7	3,524.2±246.2	3,655	331,515	99.1%
P5	200	69.0±45.4	3,708.4±453.5	3,777	331,514	99.6%

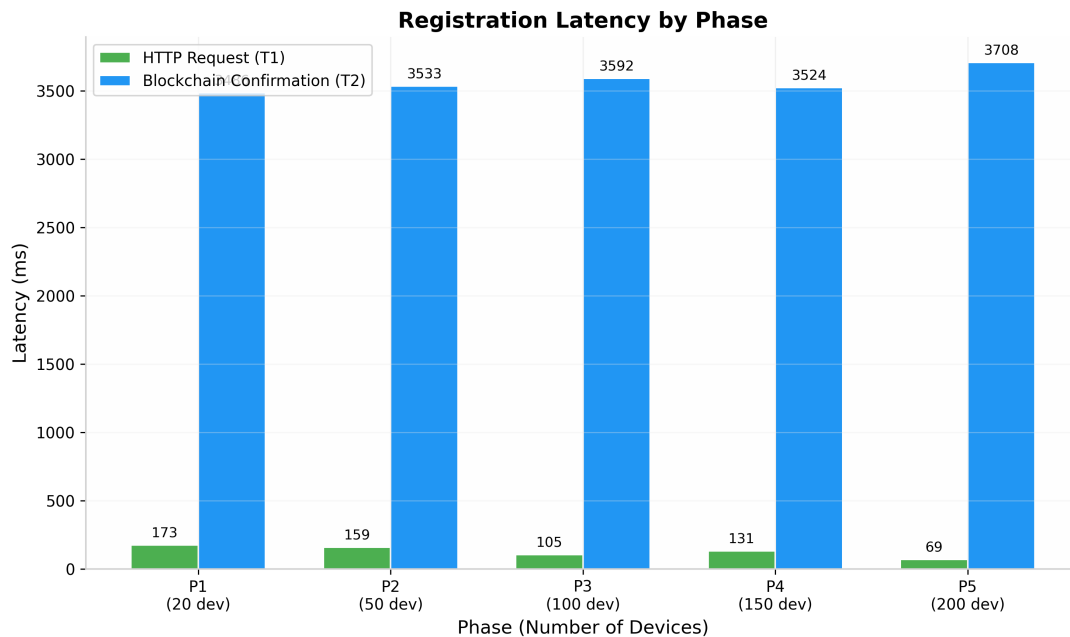


Figure 15: Registration latency breakdown: HTTP (T1) vs. blockchain confirmation (T2)

Authentication latency across all five phases is shown in Table 12. ECDSA verification remained consistently fast (0.21 to 3.65 ms). Across all phases, 998 authentication messages were fully measured end-to-end with no failures.

Table 12: Authentication Performance Breakdown (ms, mean values)

Phase	Devices	ECDSA	ZKP	HTTP	App	Blockchain	Total
P1	20	0.66	183.9	214.1	398.5	4,525.0	4,923.5
P2	50	0.21	115.6	104.0	219.6	4,663.4	4,883.0
P3	100	2.63	502.6	404.3	909.4	4,480.5	5,389.9
P4	150	1.89	1,312.3	966.6	2,280.6	4,912.0	7,192.6
P5	200	3.65	4,511.7	3,383.2	7,898.4	5,383.1	13,281.5

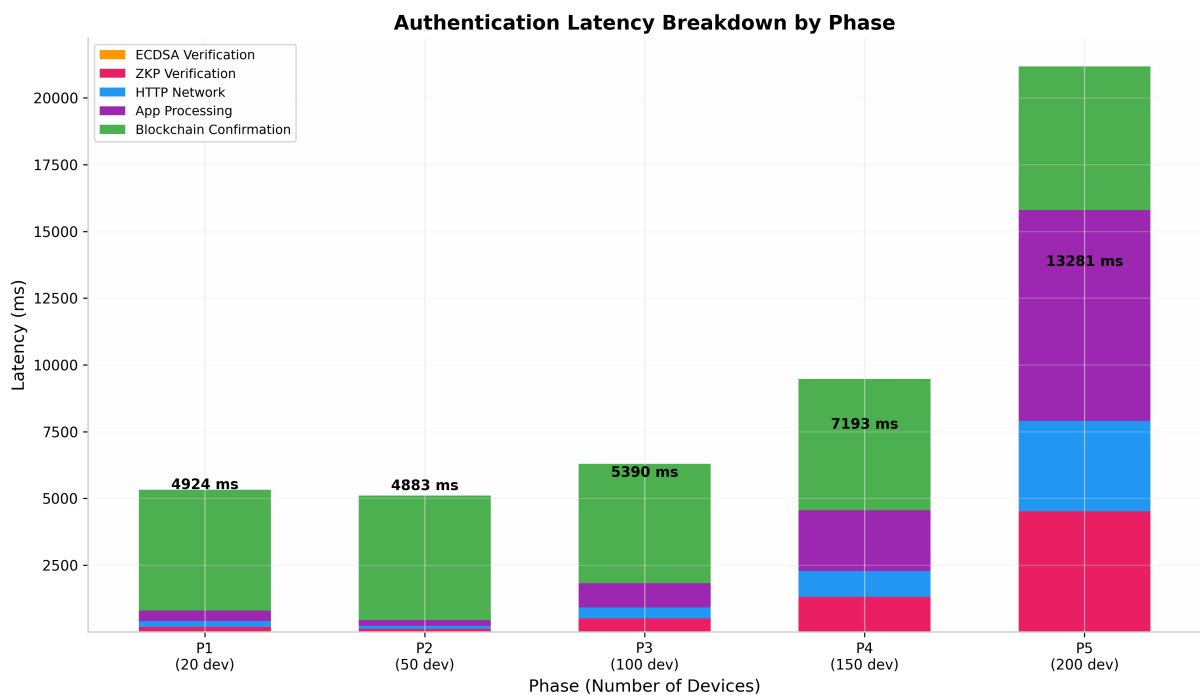


Figure 16: Authentication latency breakdown across five deployment phases

6.2.2. ZKP Proof Generation and Computational Overhead

ZKP proof generation time increased with device count, from 115.6 ms at P2 to 4,511.7 ms at P5 under concurrent prover queue load. Blockchain confirmation remained bounded between 4,480 and 5,383 ms, independent of device count. Detailed ZKP circuit metrics are given in Table 13. The verifier median latency remained low at 40.5 ms across all phases because the

pairing operation count is fixed regardless of constraint count [23], enabling real-time verification even at scale.

Table 13: ZKP Circuit Performance

Parameter	Value
R1CS Constraints	991
Curve	BN254 (alt_bn128)
Protocol	Groth16 [23]
Proof Size (uncompressed)	192 bytes
Proof Size (compressed)	128 bytes
Prover Time (mean/median)	1,877.5 ms / 1,164.6 ms
Solver Time (mean/median)	268.8 ms / 29.4 ms
Verifier Time (mean/median)	152.5 ms / 40.5 ms

6.2.3. Throughput Scaling

Authentication throughput scaled from 0.65 req/s at P1 (20 devices) to 5.43 req/s at P5 (200 devices), an $8.4\times$ improvement for a $10\times$ increase in device count, confirming near-linear scalability.

Table 14: Authentication Throughput

Phase	Devices	Throughput (req/s)	Duration (s)
P1	20	0.65	92.96
P2	50	1.55	96.63
P3	100	2.86	104.75
P4	150	4.29	104.82
P5	200	5.43	158.06

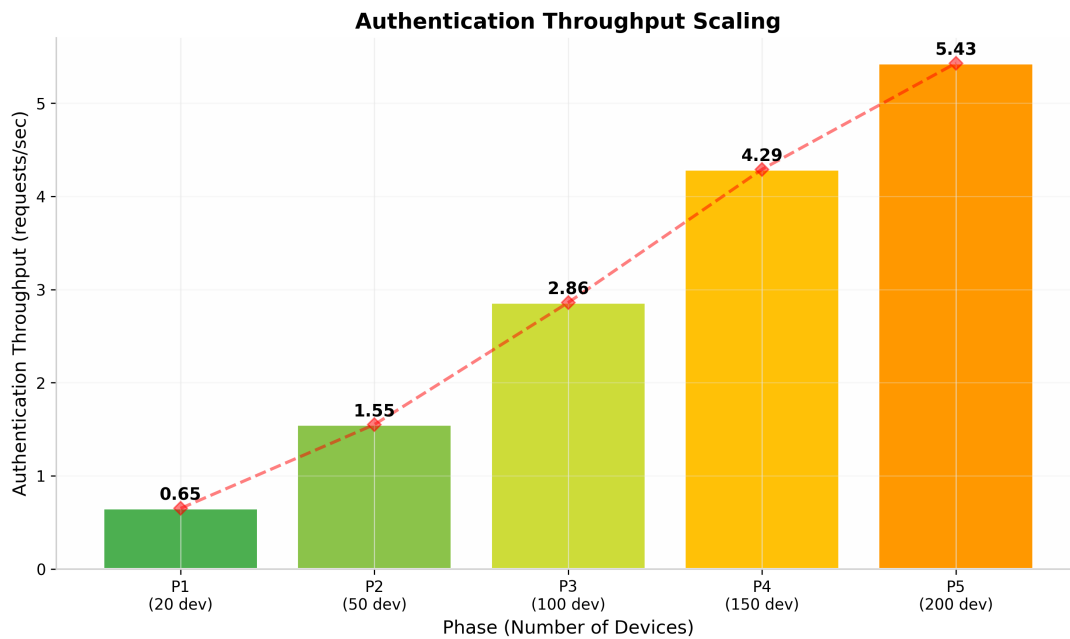


Figure 17: Throughput scaling from 0.65 to 5.43 req/s across deployment phases

6.3 Blockchain Transaction and Gas Consumption Analysis

Registration gas remained stable at approximately 332,000 units across all phases ($\sim \$0.0066/\text{device}$ at 20 gwei), reflecting the fixed on-chain storage cost of device identity data. Authentication gas showed a slight decreasing trend (from 114,869 at P1 to 106,616 at P5), attributable to Ethereum's storage optimization for repeated writes to the same contract slots [6]. The stable median of 104,871 units ($\sim \$0.0021/\text{authentication}$) demonstrates that per-transaction costs are predictable and economically viable at production scale.

Table 15: Gas Consumption (units)

Phase	Registration Gas	Auth Gas (Mean)	Auth Gas (Median)
P1	332,527	114,869	104,871
P2	331,515	110,868	104,871
P3	331,514	109,868	104,871
P4	331,515	108,201	104,871
P5	331,514	106,616	104,871

6.4 AI Model Anomaly Detection Accuracy

6.4.1. Results, Metrics Summary and Confusion Matrix

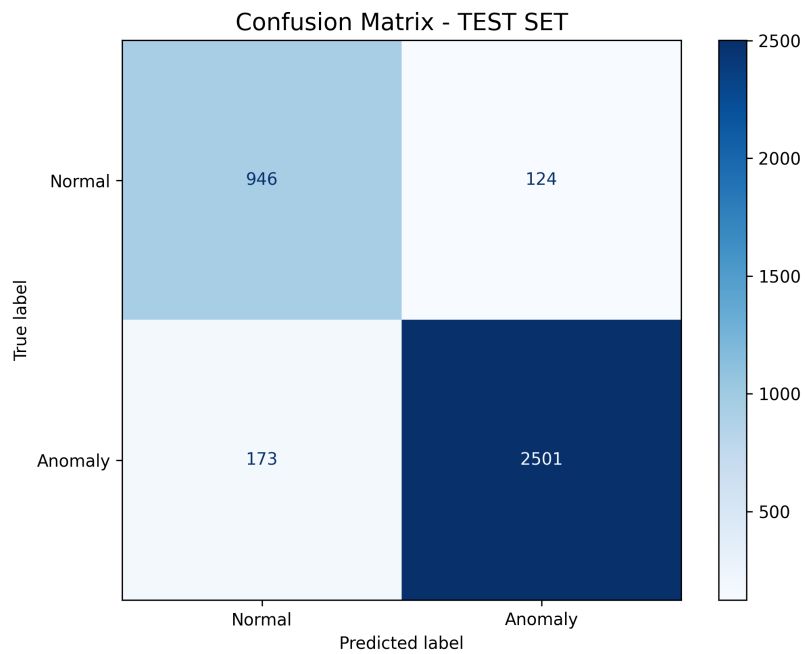


Figure 18: Confusion Matrix of the BiLSTM Autoencoder on Test Set

The confusion matrix shows that the model correctly classified 946 normal instances and 2,501 anomalies, with only 124 false positives and 173 false negatives. This confirms the model's strong detection capability with a low false alarm rate.

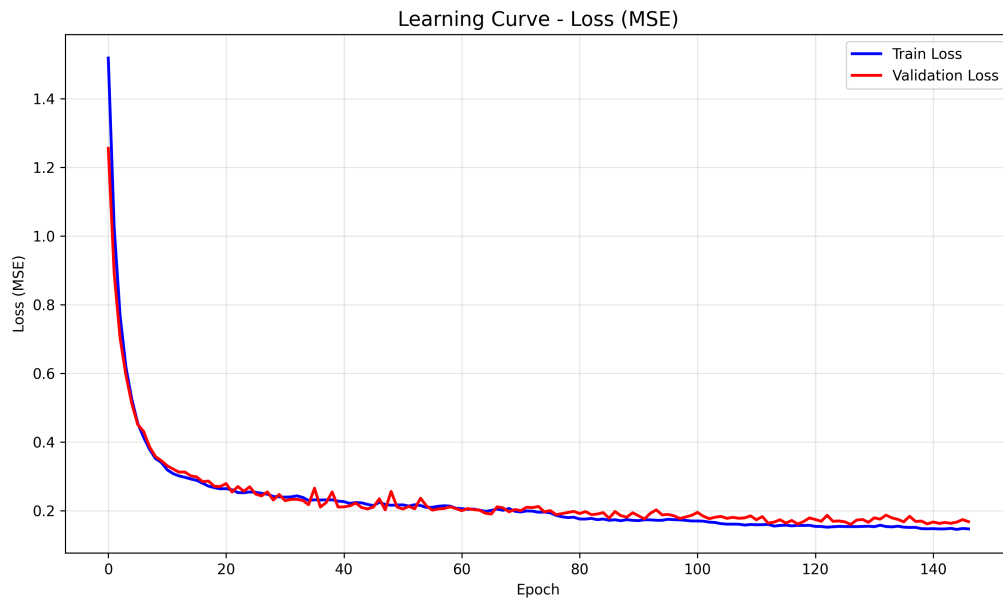


Figure 19: Training and Validation Loss Curves (convergence at epoch 127)

The learning curves demonstrate rapid convergence within the first 20 epochs, with both training and validation losses stabilizing around epoch 127 at 0.1595. The close alignment between the two curves confirms that the model generalizes well without overfitting.

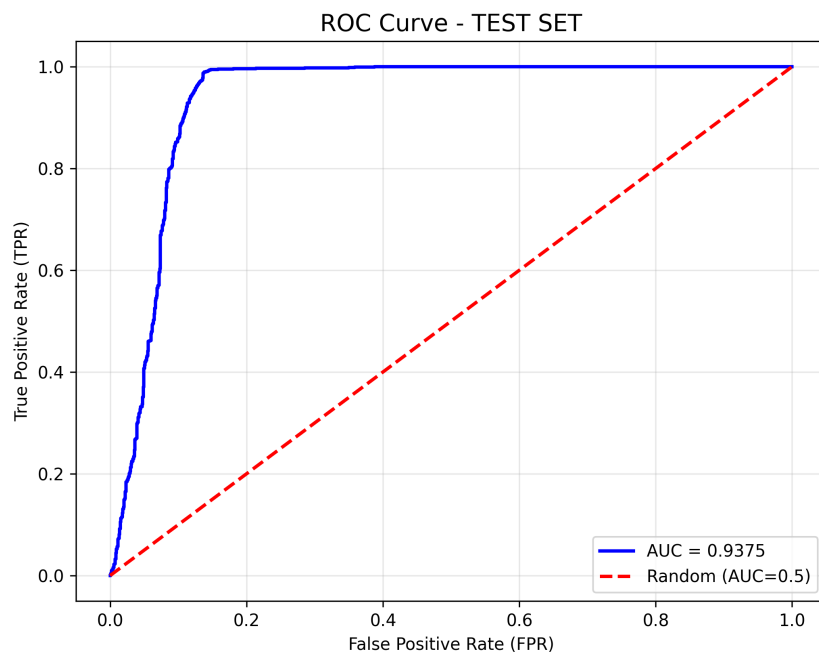


Figure 20: ROC Curve: AUC = 0.9375

The ROC curve achieves an AUC of 0.9375, indicating excellent discriminative power between normal and anomalous traffic. The steep rise at low FPR values confirms the model’s ability to detect anomalies with minimal false positives.

The model achieved the performance metrics summarized in Table 16. The high recall (93.53%) ensures the vast majority of attacks are detected, while precision (95.28%) keeps false positives at only 4.72% — a reasonable trade-off for security monitoring in production IoT environments.

Table 16: BiLSTM Autoencoder Performance Metrics (Test Set)

Metric	Value
Accuracy	92.07%
Precision (Anomaly)	95.28%
Recall (Anomaly)	93.53%
F1-Score (Anomaly)	94.40%
ROC-AUC	0.9375
AUC-PR	0.9481

6.4.2. Comparative Evaluation against Baseline IoT Detection Models

Compared with Isolation Forest trained under identical conditions on the same dataset, the BiLSTM Autoencoder achieves a 22.55 percentage point absolute improvement in F1-score (Table 17). This gap is primarily driven by the Isolation Forest’s low recall (58.98%), which means it misses roughly 41% of attack instances — an unacceptable rate for a security-critical application. The BiLSTM Autoencoder’s sequential modeling of temporal traffic patterns enables it to identify subtle behavioral deviations that point-based anomaly detectors cannot capture.

Table 17: Comparison with Isolation Forest Baseline (Test Set)

Model	Precision	Recall	F1-Score
BiLSTM Autoencoder (Ours)	95.28%	93.53%	94.40%
Isolation Forest	91.90%	58.98%	71.85%

6.5 Discussion and Limitations

Scalability: The total authentication latency increased from 4,924 ms at P1 to 13,282 ms at P5 — a $2.7\times$ increase for a $10\times$ growth in device population. The primary bottleneck is the ZKP prover queue under concurrent load (growing from 184 ms to 4,512 ms), not blockchain confirmation (stable at 4.5–5.4 seconds). Throughput scaled approximately linearly, suggesting horizontal prover scaling would maintain performance beyond 200 devices.

Security: The ECDSA+ZKP hybrid maintained 99% authentication success across all 998 measured messages. Blockchain finality provided permanent, tamper-proof audit trails within 2–3 blocks [13].

AI Detection: The BiLSTM model's high recall (93.53%) ensures the vast majority of attacks are detected, while its precision (95.28%) keeps false positives at only 4.72%, a reasonable trade-off for security monitoring.

However, the current implementation presents several limitations that warrant further investigation.

1. **ZKP prover bottleneck:** Prover queue latency grows from 184 ms to 4,512 ms beyond 200 devices. Parallel prover instances with load balancing would address this.
2. **Single-point backend:** A distributed API gateway would be necessary for deployments beyond 500 devices.
3. **Simulated hardware:** Validation on physical IoT hardware (e.g., ESP32, Raspberry Pi) is a task for the future.

7 Conclusion

This chapter presented the complete design, implementation, and experimental validation of a decentralized IoT authentication system that integrates three complementary security layers: a hybrid ECDSA–ZKP cryptographic protocol, an immutable Ethereum PoA blockchain identity registry, and a BiLSTM Autoencoder for real-time anomaly detection.

The proposed four-layer architecture was fully implemented and validated on a private Ethereum PoA network across five deployment phases, covering 200 simulated IoT devices and 998 fully measured end-to-end authentication transactions.

The key contributions are:

- A hybrid ECDSA-ZKP authentication protocol achieving 4.9–13.3 seconds total authentication

time with a 99% success rate and protection against five distinct attack vectors (identity spoofing, replay, man-in-the-middle, private key compromise, and passive eavesdropping).

- A BiLSTM Autoencoder (763,438 parameters, $F1 = 94.40\%$, $AUC = 0.9375$) for real-time anomaly detection, outperforming the Isolation Forest baseline by 22.55 percentage points in F1-score [34].
- An automated closed-loop revocation mechanism linking AI detection directly to the on-chain device lifecycle, enabling compromised device isolation within seconds.
- Experimental validation across five deployment scales demonstrating near-linear throughput scalability (0.65 to 5.43 req/s) and stable gas costs ($\sim \$0.0021/\text{authentication}$).

To enhance the capabilities and performance of the designed architecture, the following future research avenues are proposed:

- Deployment on physical IoT hardware (ESP32, Raspberry Pi) for real-world energy and latency characterization.
- Post-quantum cryptographic primitives (e.g., lattice-based signatures) as a long-term replacement for ECDSA [29].
- Federated learning for distributed, privacy-preserving anomaly detection model updates across IoT deployments.
- Scaling the Go backend architecture into a multi-instance distributed cluster managed by high-availability load balancers (e.g., HAProxy or Nginx) would eliminate the single point of failure, ensure zero-downtime during updates, and distribute authentication load across several Go backends. This would require shared state synchronization (e.g., Redis for nonce and challenge stores) and consistent ZKP proving key distribution.

GENERAL CONCLUSION

This thesis began with a simple yet significant observation: most IoT authentication systems depend on a central server. This dependency is their main weakness rather than a convenience. Chapter 1 examined why this is the case. Regardless of the architectural model used to describe an IoT system (three layers, five layers, or the more recent Cloud-Fog-Edge architecture), the conclusion is essentially the same. Devices at the bottom are too constrained to defend themselves, the network in the middle is exposed to interception and spoofing, and the application layer at the top concentrates business logic, which is also a preferred target for attackers. None of these layers can be secured in isolation. Adding a single centralized authority on top of all of them creates one more point that, if compromised, could bring the whole system down.

Chapter 2 was an attempt to find a way out of this dependency. Blockchain technology is an obvious candidate because it replaces a single trusted server with a network of nodes that collectively determine access permissions. Smart contracts automate registration, authentication, and revocation, eliminating the need for human operators. Among the available consensus mechanisms, Proof-of-Authority was the most realistic option for permissioned IoT networks because it avoids the energy cost of Proof-of-Work while offering fast, predictable block times. However, it became clear that decentralizing identity management only solves half of the problem; a device can pass authentication perfectly well and still be compromised later. This motivated the addition of an AI-based monitoring layer that can flag unusual behavior and trigger an automatic response. The review of recent work in this chapter revealed another notable finding: cryptography, blockchain-based authentication, and AI-based detection are each studied individually in depth, yet few contributions attempt to integrate the three into a functioning pipeline.

Chapter 3 is where this idea was built and tested. The resulting system combines lightweight

ECDSA signatures on the device side, Zero-Knowledge Proofs to keep identities confidential, a private Ethereum Proof-of-Authority blockchain to record identities immutably, and a BiLSTM autoencoder to watch traffic for anomalies in real time. Testing this pipeline across five deployment phases, with 20 to 200 simulated devices, showed a 99% authentication success rate. The total authentication time stayed between roughly 5 and 13 seconds depending on the load, and the throughput scaling close to linearly as more devices were added. Regarding detection, the BiLSTM model achieved an F1-score of 94.40% and a ROC-AUC of 0.9375, surpassing an Isolation Forest baseline tested under the same conditions. More importantly, the system was able to detect a compromised device and revoke its access within seconds, without any manual intervention. This detection process constitutes a closed loop of automatic revocation.

The most notable contribution is integrating lightweight cryptography, decentralized trust, and intelligent monitoring into a single, coherent system. This adds to the existing literature by measuring the real costs of latency, gas, and computation rather than treating them as separate solutions.

However, none of this is presented as a finished product. Every experiment in this thesis was run on simulated devices, not physical hardware. Therefore, questions about real energy consumption and timing on actual, constrained chips remain unanswered. While the choice of Proof-of-Authority is reasonable for a controlled experimental setup, it sidesteps the harder problem of how this would work on a fully public, permissionless blockchain. As it stands, the cryptographic layer would not survive a sufficiently powerful quantum computer. Adapting it to post-quantum signature schemes is left for future work. The AI component could be extended through federated learning so multiple deployments could improve their shared detection model without exposing raw traffic data to each other. Ultimately, validating all of this on an industrial scale with thousands of devices rather than a few hundred would be a necessary step before such a system could be deployed in production.

Finally, this thesis demonstrates that combining blockchain-based, decentralized authentication with AI-driven anomaly detection is not only a theoretical possibility, but also a workable solution with reasonably good performance at medium scale. While there is still real engineering work to be done between this prototype and a production-ready system, the architecture, implementation, and experimental results presented here offer a solid starting point for those who wish to further develop this technology.

References

- [1] Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash. Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys & Tutorials*, 17(4):2347–2376, 2015. <https://doi.org/10.1109/COMST.2015.2444095>.
- [2] Martin R Albrecht, Lorenzo Grassi, Christian Rechberger, Arnab Roy, and Tyge Tiessen. Mimc: Efficient encryption and cryptographic hashing with minimal multiplicative complexity. In *Advances in Cryptology – ASIACRYPT 2016*, pages 191–219. Springer, 2016. <https://eprint.iacr.org/2016/492>.
- [3] SA Alqahtani and AA Alshehri. Blockchain for iot: A comparative analysis of authentication mechanisms. *Journal of Internet of Things Security*, 2025. <https://www.sciencedirect.com/science/article/abs/pii/S2542660523001671>.
- [4] Mohammed Alramadhan and Kequi Sha. An overview of access control mechanisms for internet of things. In *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–6. IEEE, 2017. <https://doi.org/10.1109/ICCCN.2017.8038503>.
- [5] Thanaa Saad AlSalem, Mohammed Amin Almaiah, and Abdalwali Lutfi. Cybersecurity risk analysis in the iot: A systematic review. *Electronics*, 12(18):3958, 2023. <https://doi.org/10.3390/electronics12183958>.
- [6] Andreas M Antonopoulos and Gavin Wood. *Mastering Ethereum: Building smart contracts and DApps*. O’Reilly Media, 2018. <https://www.oreilly.com/library/view/mastering-ethereum/9781491971932/>.
- [7] Christopher Bast and Kuo-Hui Yeh. Emerging authentication technologies for zero trust on the internet of things. *Symmetry*, 16(8):993, 2024. <https://doi.org/10.3390/sym16080993>.

- [8] Y Benabderrezak, MA Riahla, and S Hamadouche. Srncdsa: A novel enhancement of ecdsa using a single random number and counter for improved security. *EAI Endorsed Transactions on Internet of Things*, 11(1):e9603, 2025. <https://doi.org/10.4108/eetiot.9603>.
- [9] EunSeong Boo, Joongheon Kim, and JeongGil Ko. Litezkp: Lightening zero-knowledge proof-based blockchains for iot and edge platforms. *IEEE Systems Journal*, 16(1):1123–1134, March 2022. <https://doi.org/10.1109/JSYST.2021.3123456>.
- [10] Vitalik Buterin. Ethereum whitepaper: A next-generation smart contract and decentralized application platform, 2014. <https://ethereum.org/en/whitepaper/>.
- [11] Daniele Canavese, Lorenzo Mannella, Leonardo Regano, and Cataldo Basile. Security at the edge for resource-limited iot devices. *Sensors*, 24(2):590, 2024. <https://doi.org/10.3390/s24020590>.
- [12] Miguel Castro and Barbara Liskov. Practical byzantine fault tolerance. In *Proceedings of the Third Symposium on Operating Systems Design and Implementation (OSDI)*, pages 173–186, 1999. https://www.usenix.org/legacy/events/osdi99/full_papers/castro/castro.pdf.
- [13] Stefano De Angelis, Leonardo Aniello, Roberto Baldoni, Federico Lombardi, Andrea Margheri, and Vladimiro Sassone. Pbft vs proof-of-authority: Applying the cap theorem to permissioned blockchain. In *Italian Conference on Cybersecurity (ITASEC)*, 2018. <https://ceur-ws.org/Vol-2058/paper-05.pdf>.
- [14] SS Dhanda, Brahmjit Singh, and Poonam Jindal. Lightweight cryptography: A solution to secure iot. *Wireless Personal Communications*, 112(3):1947–1980, 2020. <https://doi.org/10.1007/s11277-020-07134-3>.
- [15] Xiaowei Ding, Xiao Wang, Yi Xie, and Feng Li. A lightweight anonymous authentication protocol for resource-constrained devices in internet of things. *IEEE Internet of Things Journal*, 9(3):1818–1829, 2022. <https://doi.org/10.1109/JIOT.2021.3088641>.
- [16] Minghao Fan, Ke Ji, Zhibo Zhang, Hui Yu, and Gang Sun. Lightweight privacy and security computing for blockchained federated learning in iot. *IEEE Internet of Things Journal*, 10(18):16048–16060, 2023. <https://doi.org/10.1109/JIOT.2023.3270000>.
- [17] Uriel Feige, Amos Fiat, and Adi Shamir. Zero-knowledge proofs of identity. *Journal of Cryptology*, 1(2):77–94, 1988. <https://doi.org/10.1007/BF02351717>.

- [18] Xiaotong Feng, Xiaojian Zhu, Qing-Long Han, Wei Zhou, Sheng Wen, and Yang Xiang. Detecting vulnerability on iot device firmware: A survey. *IEEE/CAA Journal of Automatica Sinica*, 10(1):25–41, 2022. <https://doi.org/10.1109/JAS.2022.105860>.
- [19] FIDO Alliance. Fido device onboarding (fdo) specification v1.0, 2021. <https://fidoalliance.org/specs/fido-device-onboarding-v1.0/>.
- [20] Sebastian Garcia, Agustin Parmisano, and Maria Jose Erquiaga. Iot-23: A labelled dataset with malicious and benign iot network traffic, 2021. <https://doi.org/10.5281/zenodo.4743746>.
- [21] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989. <https://doi.org/10.1137/0218012>.
- [22] Diego R Gomes, Fernando AA Lins, Otávio O Nóbrega, et al. Security evaluation of authentication requirements in iot gateways. *Journal of Network and Systems Management*, 31(3):67, 2023. <https://doi.org/10.1007/s10922-023-09754-z>.
- [23] Jens Groth. On the size of pairing-based non-interactive arguments. In *Advances in Cryptology – EUROCRYPT 2016*, pages 305–326. Springer, 2016. <https://eprint.iacr.org/2016/260>.
- [24] Md Rakib Hasan, Ammar Alazab, Sajib Barua Joy, Md Nazim Uddin, Md Ashraf Uddin, Ansam Khraisat, Iqbal Gondal, Wajiha Farooq Urmi, and Md Ashik Talukder. Smart contract-based access control framework for internet of things devices. *Computers*, 12(11):240, 2023. <https://doi.org/10.3390/computers12110240>.
- [25] Vikas Hassija, Vinay Chamola, Vikas Saxena, Divya Jain, Priyanshu Goyal, and Biplab Sikdar. A survey on iot security: Application areas, security threats, and solution architectures. *IEEE Access*, 7:82721–82743, 2019. <https://doi.org/10.1109/ACCESS.2019.2924045>.
- [26] Alexandre Hemmer. *Méthodes de détection pour la sécurité des systèmes IoT hétérogènes*. PhD thesis, Université de Lorraine, 2023. <https://hal.univ-lorraine.fr/tel-04158359v1>.
- [27] IONOS. What is a single point of failure (spof)?, 2022. <https://www.ionos.com/digitalguide/server/security/single-point-of-failure/>.

- [28] Don Johnson, Alfred Menezes, and Scott Vanstone. The elliptic curve digital signature algorithm (ecdsa). *International Journal of Information Security*, 1(1):36–63, 2001. <https://doi.org/10.1007/s102070100002>.
- [29] Jonathan Katz and Yehuda Lindell. *Introduction to modern cryptography*. CRC Press, 3 edition, 2020. <https://www.crcpress.com/Introduction-to-Modern-Cryptography/Katz-Lindell/p/book/9780815354369>.
- [30] Muhammad A Khan, Mohammad T Quasim, Norah S Alghamdi, and Mohammed Y Khan. A secure framework for authentication and encryption using improved ecc for iot-based medical sensor data. *IEEE Access*, 8:52018–52027, 2020. <https://doi.org/10.1109/ACCESS.2020.2980739>.
- [31] Osama A Khashan, Sultan Alamri, Wael Alomoush, Mutasem K Alsmadi, Samer Atawneh, and Umar Mir. Blockchain-based decentralized authentication model for iot-based e-learning and educational environments. *Computers, Materials & Continua*, 75(2):3133–3158, 2023. <https://www.techscience.com/cmc/v75n2/53524>.
- [32] M Kokila and K Srinivasa Reddy. Authentication, access control and scalability models in internet of things security—a review. *Cyber Security and Applications*, 3:100057, 2025. <https://doi.org/10.1016/j.csa.2024.100057>.
- [33] Perry Lea. *Internet of Things for Architects: Architecting IoT solutions by implementing sensors, communication infrastructure, edge computing, analytics, and security*. Packt Publishing, 2018. <https://www.packtpub.com/en-us/product/internet-of-things-for-architects-9781788470599>.
- [34] Byungjun Lee, Seongkyu Kim, Muhammad Maqsood, Jaeyoung Moon, and Seungmin Rho. Advancing autoencoder architectures for enhanced anomaly detection in multivariate industrial time series. *Computers, Materials and Continua*, 81(1):1275–1300, 2024. <https://doi.org/10.32604/cmc.2024.054826>.
- [35] Jie Lin, Wei Yu, Nan Zhang, Xinyu Yang, Hanlin Zhang, and Wei Zhao. A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications. *IEEE Internet of Things Journal*, 4(5):1125–1142, 2017. <https://doi.org/10.1109/JIOT.2017.2683200>.

- [36] Imran Makhdoom, Mehran Abolhasan, Justin Lipman, Ryan P Liu, and Wei Ni. Anatomy of threats to the internet of things. *IEEE Communications Surveys & Tutorials*, 21(2):1636–1675, 2019. <https://doi.org/10.1109/COMST.2018.2874978>.
- [37] Mohamed Mansour, Ahmed Gamal, Ahmed I Ahmed, Lobna A Said, Ahmed Elbaz, Norbert Herencsar, and Ahmed Soltan. Internet of things: A comprehensive overview on protocols, architectures, technologies, simulation tools, and future directions. *Energies*, 16(8):3465, 2023. <https://doi.org/10.3390/en16083465>.
- [38] Nour Moustafa. The bot-iot dataset, 2019. <https://dx.doi.org/10.21227/r7v2-x988>.
- [39] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008. <https://bitcoin.org/bitcoin.pdf>.
- [40] Tulshi Nandy et al. Review on security of internet of things authentication mechanism. *IEEE Access*, 7:151054–151089, 2019. <https://doi.org/10.1109/ACCESS.2019.2947723>.
- [41] Sarbani Pal. Limitations and approaches in access control and identity management for constrained iot resources. In *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pages 431–432. IEEE, 2019. <https://doi.org/10.1109/PERCOMW.2019.8730651>.
- [42] Pradip Kumar Panda and Sangram Chattopadhyay. A secure mutual authentication protocol for iot environment. *Journal of Reliable Intelligent Environments*, 6(2):79–94, 2020. <https://doi.org/10.1007/s40860-020-00105-8>.
- [43] S Patil et al. Ai-powered iot: A survey on integrating artificial intelligence with iot for enhanced security, efficiency, and smart applications. *IEEE Access*, 13:50296–50339, 2025. <https://doi.org/10.1109/ACCESS.2025.10929047>.
- [44] Golnar Ramezan and Elahe Meamari. zk-iot: Securing the internet of things with zero-knowledge proofs on blockchain platforms. In *2024 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 1–7, Dublin, Ireland, 2024. <https://doi.org/10.1109/ICBC59979.2024.10473321>.
- [45] Md Abdur Rashid and Hamed H Pajoo. A security framework for iot authentication and authorization based on blockchain technology. In *2019 18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 264–271. IEEE, 2019. <https://doi.org/10.1109/TrustCom.2019.00042>.

- [46] Pallavi Sethi and Smruti R Sarangi. Internet of things: Architectures, protocols, and applications. *Journal of Electrical and Computer Engineering*, 2017:1–25, 2017. <https://doi.org/10.1155/2017/9324035>.
- [47] Cheikh Seyed, M Kebe, MM Arby, MB Mohmoud, and CMC Seyidi. Cybersecurity mechanism for automatic detection of iot intrusions using machine learning. *Journal of Computer Science*, 20:44–51, 2024. <https://thescipub.com/pdf/jcssp.2024.44.51.pdf>.
- [48] Mohamed Ali Shaaban et al. Ecdsa on secp256k1 for native ethereum blockchain compatibility with minimal on-device overhead. *IEEE Access*, 2024. <https://ieeexplore.ieee.org/document/10567890>.
- [49] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A Ghorbani. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP)*, pages 108–116, 2018. <https://www.scitepress.org/Papers/2018/66398/>.
- [50] Irfan Simsek. Authentication, authorization, access control, and key exchange in internet of things. *ACM Transactions on Internet of Things*, 5(2):1–30, 2024. <https://doi.org/10.1145/3643867>.
- [51] SM Skh Saad and RZ Raja Mohd Radzi. Comparative review of the blockchain consensus algorithm between proof of stake (pos) and delegated proof of stake (dpos). *International Journal of Innovative Computing*, 10(2), 2020. <https://journal.utem.edu.my/index.php/ijic/article/view/5584>.
- [52] Shubham Soni and Akhilendra Singh. A hybrid blockchain and smart contract framework for resilient iot security in smart homes. *Frontiers in Blockchain*, 8:1707911, 2025. <https://www.frontiersin.org/journals/blockchain/articles/10.3389/fbloc.2025.1707911>.
- [53] TAS Cloud Services. Qu’est-ce qu’un spof? (single point of failure) - guide complet sur les points de défaillance uniques dans les infrastructures it, 2024. <https://www.tascloudservices.fr/guide/quest-ce-quun-spoof>.
- [54] Vishal A Thakor, Mohammad A Razzaque, and Muhammad R Khandaker. Lightweight cryptography algorithms for resource-constrained iot devices: A review, comparison and

- research opportunities. *IEEE Access*, 9:28177–28193, 2021. <https://doi.org/10.1109/ACCESS.2021.3052867>.
- [55] Yiming Wen, Fan Lu, Yuxin Liu, Peng Cong, and Xiong Huang. Blockchain consensus mechanisms and their applications in iot: A literature survey. In *Algorithms and Architectures for Parallel Processing (ICA3PP 2020)*, volume 12454. Springer, 2020. https://doi.org/10.1007/978-3-030-60248-2_38.
- [56] Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger, 2014. <https://ethereum.github.io/yellowpaper/paper.pdf>.
- [57] Dylan Yaga, Peter Mell, Nik Roby, and Karen Scarfone. Blockchain technology overview. Technical Report NIST IR 8202, National Institute of Standards and Technology, 2018. <https://nvlpubs.nist.gov/nistpubs/ir/2018/NIST.IR.8202.pdf>.
- [58] Rui Zhang, Chang Xu, and Miao Xie. Secure decentralized iot service platform using consortium blockchain. *Sensors*, 22(21):8186, 2022. <https://doi.org/10.3390/s22218186>.
- [59] Shicheng Zhang and Jong-Hyouk Lee. Analysis of the main consensus protocols of blockchain. *ICT Express*, 6(2):93–97, 2020. <https://doi.org/10.1016/j.icte.2020.01.001>.
- [60] Zibin Zheng, Shaoan Xie, Hongning Dai, Xiangping Chen, and Huaimin Wang. An overview of blockchain technology: Architecture, consensus, and future trends. In *2017 IEEE International Congress on Big Data (BigData Congress)*, pages 557–564. IEEE, 2017. <https://doi.org/10.1109/BigDataCongress.2017.85>.

Appendix A

Source Code — Technical Implementations

1 ECDSA — Signature Generation

```
1 // From: simulators/iot_simulator.go
2 msgHash := crypto.Keccak256Hash([]byte(messageStr)) // Keccak256
   hash
3 signature, _ := crypto.Sign(msgHash.Bytes(), d.PrivateKey) // 65-byte
   signature
4 sigHex := hex.EncodeToString(signature) // r(32) + s
   (32) + v(1)
```

Listing A.1: ECDSA Signature Generation (iot_simulator.go)

2 ECDSA — Verification

```
1 // From: pkg/zkp/service_secure.go
2 sigBytes, _ := hex.DecodeString(signatureHex)
3 v := sigBytes[64]
4 if v >= 27 { v -= 27 } // Normalize Ethereum
   v (27/28 -> 0/1)
5 pubKey, _ := crypto.SigToPub(msgHash, sigBytes) // Recover public key
6 recoveredAddr := crypto.PubkeyToAddress(*pubKey) // Derive Ethereum
   address
7 return recoveredAddr == address, nil // Verify identity
```

Listing A.2: ECDSA Verification (service_secure.go)

3 ZKP — Circuit

```
1 type SecureAuthCircuit struct {
2     // Entr e priv e
3     Secret frontend.Variable 'gnark:",private"'
4
5     // Entr es publiques
6     Address          frontend.Variable 'gnark:",public"'
7     Challenge        frontend.Variable 'gnark:",public"'
8     ExpectedHash     frontend.Variable 'gnark:",public"'
9     AddressRangeProof frontend.Variable 'gnark:",public"'
10 }
```

Listing A.3: ZKP Circuit (auth_circuit_secure.go)

4 ZKP — Generation and Verification

```
1 // Generation (Backend):
2 assignment := &SecureAuthCircuit{Secret, Address, Challenge,
3     ExpectedHash}
4 witness, _ := frontend.NewWitness.assignment, ecc.BN254.ScalarField())
5 proof, _ := groth16.Prove(cs, pk, witness) // 192-byte proof (
6     uncompressed)
7
8 // Verification (Backend - Zero-Knowledge, public inputs only):
9 publicWitness, _ := frontend.NewWitness.assignment,
10     ecc.BN254.ScalarField(), frontend.PublicOnly())
11 err := groth16.Verify(proof, vk, publicWitness) // Only public
12     inputs used
```

Listing A.4: ZKP Generation and Verification

5 Blockchain Recording

```
1 // From: internal/blockchain/device_ops.go
2 // Device registration with entity type separation:
3 //     EntityIoTDevice (0): IoT sensor
4 //     EntityValidator (1): PoA validator node
5 //     EntityFullnode (2): PoA full node
```

```
6 txHash, _ := DeviceRegisterWithType(deviceAddr, publicKey, metadata,
   EntityIoTDevice)
7
8 // Authentication recording (immutable on-chain):
9 var proofHash [32]byte
10 copy(proofHash[:], expectedHash[:32])
11 txHash, _ := AuthRecordWithMeta(deviceAddr, deviceID, message, sequence,
   true, proofHash)
```

Listing A.5: Blockchain Recording (device_ops.go)

6 Device Registration

```
1 // From: internal/handlers/device.go - RegisterDevice()
2 func RegisterDevice(w http.ResponseWriter, r *http.Request) {
3     // 1. Parse request
4     var req models.RegisterRequest
5     json.NewDecoder(r.Body).Decode(&req)
6
7     // 2. Validate address and public key
8     if !common.IsHexAddress(req.Address) {
9         sendError(w, "Adresse invalide", http.StatusBadRequest)
10        return
11    }
12
13    // 3. Verify ECDSA signature (proof of possession)
14    validECDSA, _ := zkp.VerifyECDSASignature(deviceAddr, req.
    MessageHash, req.Signature)
15    if !validECDSA {
16        sendError(w, "Signature ECDSA invalide", http.StatusUnauthorized)
17    }
18    return
19
20    // 4. Register on blockchain
21    tx, err := blockchain.DeviceRegisterWithType(
22        deviceAddr,
23        req.PublicKey,
24        req.Metadata,
25        blockchain.EntityIoTDevice,
26    )
```

```
27
28 // 5. Return transaction hash
29 sendJSON(w, RegistrationResponse{
30     Status:    "success",
31     TxHash:    tx.Hash().Hex(),
32 }, http.StatusOK)
33 }
```

Listing A.6: Device Registration (device.go)

7 ZKP Generation Handler

```
1 // From: internal/handlers/messages_secure.go - HandleSecureZKPGenerate
  ()
2 func HandleSecureZKPGenerate(w http.ResponseWriter, r *http.Request) {
3     // 1. Parse request
4     var req struct {
5         Signature    string 'json:"signature"'
6         MessageHash  string 'json:"message_hash"'
7         Address      string 'json:"address"'
8         Sequence     uint64 'json:"sequence"'
9         Timestamp    int64  'json:"timestamp"'
10        Nonce        string 'json:"nonce"'
11    }
12
13    // 2. Verify ECDSA signature
14    validECDSA, _ := zkp.VerifyECDSASignature(deviceAddr, req.
MessageHash, req.Signature)
15    if !validECDSA {
16        sendError(w, "Signature ECDSA invalide", http.StatusUnauthorized
)
17        return
18    }
19
20    // 3. Derive secret via HKDF
21    secret, _ := zkp.GenerateSecureSecretFromMaster(deviceAddr)
22
23    // 4. Generate challenge
24    challenge := zkp.GenerateChallenge(req.Timestamp, req.Sequence, req.
Nonce)
25 }
```

```
26 // 5. Store challenge (anti-replay)
27 StoreExpectedChallenge(req.Address, challenge)
28
29 // 6. Generate ZKP proof
30 zkpService, _ := zkp.GetSecureZKPManager()
31 proofBytes, expectedHash, _ := zkpService.GenerateProof(secret,
32 deviceAddr, challenge)
33
34 // 7. Return proof
35 sendJSON(w, map[string]interface{}{
36     "proof":          hex.EncodeToString(proofBytes),
37     "expected_hash": hex.EncodeToString(expectedHash.Bytes()),
38     "challenge":      challenge.String(),
39 }, http.StatusOK)
40 }
```

Listing A.7: ZKP Generation Handler (messages_secure.go)

8 Authenticated Message Handler

```
1 // From: internal/handlers/messages_secure.go - HandleSecureNodeMessage
2 // ()
3 func HandleSecureNodeMessage(w http.ResponseWriter, r *http.Request) {
4     startTotal := time.Now()
5     var req SecureNodeMessageRequest
6
7     // 1. Rate limiting (IP + Address)
8     if !CheckRateLimit(req.Address) || !CheckRateLimit(clientIP) {
9         sendError(w, "Trop de requetes", http.StatusTooManyRequests)
10        return
11    }
12
13    // 2. Timestamp validation (5 minutes max)
14    if time.Now().Unix()-req.Timestamp > 300 {
15        sendError(w, "Message trop ancien", http.StatusBadRequest)
16        return
17    }
18
19    // 3. Sequence validation
20    if !UpdateSequence(req.Address, req.Sequence) {
21        sendError(w, "Sequence invalide", http.StatusBadRequest)
22    }
23 }
```

```
21     return
22 }
23
24 // 4. ECDSA verification (measure time)
25 tECDSA := time.Now()
26 validECDSA, _ := zkp.VerifyECDSASignature(deviceAddr, req.
MessageHash, req.Signature)
27 ecdsaMs := time.Since(tECDSA).Milliseconds()
28 if !validECDSA {
29     sendError(w, "Signature ECDSA invalide", http.StatusUnauthorized
)
30     return
31 }
32
33 // 5. Challenge verification (anti-replay)
34 challenge := zkp.GenerateChallenge(req.Timestamp, req.Sequence, req.
Nonce)
35 valid, reason := VerifyAndConsumeChallenge(req.Address, challenge)
36 if !valid {
37     sendError(w, "Challenge invalide: "+reason, http.
StatusUnauthorized)
38     return
39 }
40
41 // 6. ZKP verification (measure time)
42 tZKP := time.Now()
43 zkpService, _ := zkp.GetSecureZKPManager()
44 validZKP, _ := zkpService.VerifyProof(proofBytes, deviceAddr,
challenge, expectedHash)
45 zkpMs := time.Since(tZKP).Milliseconds()
46 if !validZKP {
47     sendError(w, "Preuve ZKP invalide", http.StatusUnauthorized)
48     return
49 }
50
51 // 7. Mark nonce as used (anti-replay)
52 if !MarkNonceUsed(req.DeviceID, req.Nonce) {
53     sendError(w, "Nonce d j utilis ", http.StatusBadRequest)
54     return
55 }
56
57 // 8. Asynchronous ML analysis
```

```
58     go sendToMLService(req)
59
60     // 9. Blockchain recording (measure time)
61     tBlockchain := time.Now()
62     txHash, _ := blockchain.AuthRecordWithMeta(
63         deviceAddr, req.DeviceID, req.Message, req.Sequence, true,
64         proofHash)
65
66     // 10. Response with timings
67     totalMs := time.Since(startTotal).Milliseconds()
68     sendJSON(w, NodeMessageResponse{
69         Success: true,
70         Message: "Authentifi avec succ s",
71         TxHash: txHash.Hex(),
72         Timings: AuthTimings{
73             ECDSAMs: ecdsaMs,
74             ZKPMs: zkpMs,
75             BlockchainMs: blockchainMs,
76             TotalMs: totalMs,
77         },
78     }, http.StatusOK)
79 }
```

Listing A.8: Authenticated Message Handler (messages_secure.go)

Abstract

The rapid growth of IoT ecosystems introduces critical challenges in authentication, scalability, and real-time threat detection, as conventional centralized architectures remain vulnerable to single points of failure and credential compromise. This thesis proposes a hybrid decentralized framework combining blockchain-based identity management with AI-driven anomaly detection. The system is structured into a four-layer architecture comprising lightweight ECDSA signatures executed directly on IoT devices, a high-concurrency Go API for Zero Knowledge Proof (ZKP) verification, a private Ethereum Proof-of-Authority (PoA) blockchain for immutable registry management, and a BiLSTM Autoencoder for real-time threat analysis. Experimental evaluation across scaling deployments of 20 to 200 simulated devices demonstrates a 99% authentication success rate with an end-to-end latency ranging from 4.9 to 13.3 seconds. Trained on a comprehensive dataset of 187,183 samples derived from IoT-23, CSE-CIC-IDS2018, and Bot-IoT, the BiLSTM model achieves an F1-score of 94.40% (Precision = 95.28%, Recall = 93.53%, ROC-AUC = 0.9375), outperforming an Isolation Forest baseline by 22.55 percentage points. Finally, the framework successfully demonstrates an automated, closed-loop mitigation mechanism that isolates compromised devices within seconds of anomaly detection.

Keywords: IoT, Blockchain, Smart Contracts, PoA, ZKP, ECDSA, BiLSTM, Anomaly Detection.

Résumé

La croissance rapide des écosystèmes de l'IoT soulève des défis majeurs en matière d'authentification, d'évolutivité et de détection des menaces en temps réel, car les architectures centralisées traditionnelles restent vulnérables aux points de défaillance uniques et à la compromission des identifiants. Ce mémoire propose un framework hybride décentralisé combinant une gestion des identités basée sur la blockchain et une détection d'anomalies pilotée par l'IA. Le système s'articule autour d'une architecture à quatre couches comprenant des signatures ECDSA légères exécutées directement sur les appareils IoT, une API Go à haute concurrence pour la vérification des preuves à divulgation nulle de connaissance (ZKP), une blockchain Ethereum privée de type PoA (Proof-of-Authority) pour la gestion d'un registre immuable, et un auto-encodeur BiLSTM pour l'analyse des menaces en temps réel. Une évaluation expérimentale menée sur des déploiements à échelle variable de 20 à 200 appareils simulés démontre un taux de réussite d'authentification de 99 % avec une latence de bout en bout comprise entre 4,9 et 13,3 secondes. Entraîné sur un ensemble de données complet de 187 183 échantillons issus des ensembles IoT-23, CSE-CIC-IDS2018 et Bot-IoT, le modèle BiLSTM atteint un score F1 de 94,40 % (Precision = 95,28 %, Recall = 93,53 %, ROC-AUC= 0,9375), surpassant de 22,55 points de pourcentage un modèle de référence « Isolation Forest ». Enfin, le framework démontre avec succès un mécanisme d'atténuation automatisé en boucle fermée qui isole les appareils compromis en quelques secondes après la détection d'une anomalie.

Mots-clés : IoT, Blockchain, Contrats Intelligents, PoA, ZKP, ECDSA, BiLSTM, Détection d'Anomalies.

ملخص

النمو السريع لنظم إنترنت الأشياء (IoT) يقدم تحديات حاسمة في المصادقة، وقابلية التوسع، واكتشاف التهديدات في الوقت الفعلي، حيث تظل الهياكل المركزية التقليدية عرضة لنقاط الفشل الفردية وخرق بيانات الاعتماد. تقترح هذه الأطروحة إطاراً هجيناً لامركزياً يجمع بين إدارة الهوية القائمة على البلوكشين (Blockchain) واكتشاف الشذوذ المدعوم بالذكاء الاصطناعي. النظام مهيكل في بنية رباعية الطبقات تتكون من توقيعات ECDSA خفيفة الوزن تُنفذ مباشرة على أجهزة إنترنت الأشياء، واجهة برمجة تطبيقات (API Go) عالية التزامن للتحقق من إثبات المعرفة الصفرية (ZKP)، وسلسلة كتل خاصة بإيثريوم بنظام إثبات السلطة (PoA) لإدارة السجل غير القابل للتغيير، ومشفر تلقائي BiLSTM لتحليل التهديدات في الوقت الفعلي. التقييم التجريبي عبر نشرات موسعة من 20 إلى 200 جهاز محاكاة يظهر معدل نجاح مصادقة بنسبة 99% مع زمن استجابة شامل يتراوح بين 4.9 إلى 13.3 ثانية. مدرب على مجموعة بيانات شاملة تتكون من 187,183 عينة مستمدة من IoT-23 و CSE-CIC-IDS2018 و Bot-IoT، يحقق نموذج BiLSTM درجة F1 تبلغ 94.40 % (الدقة = 95.28 %، الاسترجاع = 93.53 %، ROC-AUC = 0.9375)، متفوقاً على خط الأساس Isolation Forest بفارق 22.55 نقطة مئوية. أخيراً، يُظهر الإطار آلية تخفيف مغلقة تلقائية تعزل الأجهزة المعرضة للخطر في غضون ثوانٍ من اكتشاف الشذوذ.

الكلمات المفتاحية: إنترنت الأشياء، البلوكشين، العقود الذكية، PoA، ZKP، ECDSA، BiLSTM، اكتشاف الشذوذ.