Université de Strasbourg

École doctorale MSII

Building a Secure and Scalable Distributed Network using Blockchain and Zero Trust for IIoT

THÈSE

pour obtenir le grade de

Doctorat de l'Université de Strasbourg

(Informatique)

Soutenue le / Defended on : 17 juillet 2025

présentée par

Fatemeh STODT

Composition du jury

Directeurs de thèse:	Prof. Dr. **Christoph REICH**, Full Professor, Institute for Data Science, Cloud Computing & IT Security (IDACUS), Furtwangen University, Germany
	Dr. **Fabrice THEOLEYRE**, Directeur de recherche, CNRS, France
Rapporteurs:	Prof. **Lyes KHOUKI**, Full Professor, ENSICAEN, Université de Normandie
	Dr. **Valeria LOSCRI**, Directrice de recherche, Inria, France
Examinateurs:	Prof. **Isabelle CHRISMENT**, Full Professor, Telecom Nancy, Université de Lorraine

Laboratoire des sciences de l'ingénieur, de l'informatique et de l'imagerie
ICUBE — UMR 7357

Acknowledgments

To the Holy Trinity – Father, Son, and Holy Spirit – who enlighten my life.

I wish to express my deep gratitude to Father John, Gerontissa and my sisters in Chrysopigi Monastery of Crete, whose spiritual support has uplifted both my soul and my scientific journey.

I am especially thankful to my beloved husband Jan, who stood by me through the most difficult times of our lives with unwavering support and love. Your patience and strength have meant more than words can express.

My heartfelt thanks go to my parents, Roland and Ursula, who welcomed me into their family with open arms and endless kindness. I also wish to thank Anna Krinis for her continued mental and emotional support throughout these past years.

I extend my sincere appreciation to my supervisors, Fabrice Theoleyre and Christoph Reich, for their invaluable guidance, patience, and encouragement. Their mentorship has been instrumental in shaping this work.

Fatemeh Stodt,
April 2025

Abstract

These days, Industrial Internet of Things (IIoT) systems are widely used in the transportation, industrial, energy, and healthcare industries. They empower real-time data collection, autonomous decision-making, and seamless connectivity. However, the increased scale of IIoT deployments and their mission-critical nature introduce significant security and scalability challenges. Traditional perimeter-based security approaches are not sufficient to cope with sophisticated attacks and highly heterogeneous device ecosystems.

This thesis addresses the need for a robust, scalable, and adaptive security framework that ensures trustworthiness and reliability in large-scale, distributed IIoT environments. We suggest a unique combination of distributed approach and dynamic Zero Trust Architecture (ZTA) principles to safeguard device IDs, provide access control, and immediately identify irregularities. Our approach integrates advanced Identity Management (IdM) protocols, dynamic policy enforcement, and context-aware anomaly detection models tailored for industrial processes. Additionally, we introduce a lightweight blockchain solution to provide a tamper-proof record of critical events and secure cross-domain collaboration key elements for enabling distributed trust among multiple industrial stakeholders.

Through simulations and a real-world testbed, we demonstrate that our framework offers enhanced security against both known and zero-day attacks, maintains high throughput even in resource-constrained scenarios, and adapts to changes in the network environment and threat landscape. Our findings underline the feasibility of harmonizing ZTA concepts with blockchain technology to future-proof IIoT systems against evolving operational and security demands. We believe this combined methodology can become a security layer for next-generation industrial networks, improving both their resilience to cyber threats and their ability to scale seamlessly as Industry 4.0 continues to evolve.

List of Publications

Journal Papers

P1. Fatemeh Stodt and Christoph Reich. "Bridge of Trust: Cross Domain Authentication for Industrial Internet of Things (IIoT) Blockchain over Transport Layer Security (TLS)". in: ***Electronics*** 12.11 (2023), p. 2401.

P2. Fatemeh Stodt and Christoph Reich. "Digital Wallets and Identity Management: Pioneering Advances for Cloud Service Evolution". In: ***International Journal on Advances in Software*** 17.1 (2024), pp. 13–22.

P3. Fatemeh Stodt, Mohammed BM Kamel, Christoph Reich, Fabrice Theoleyre, and Peter Ligeti. "Blockchain-Based Privacy-Preserving Shop Floor Auditing Architecture". In: ***IEEE Access*** 12 (2024), pp. 26747–26758.

P4. Fatemeh Stodt, Mohammed Alshawki, Christoph Reich, Peter Ligeti, and Fabrice Theoleyre. "Securing the Future: Lightweight Blockchain Solutions for IIoT and IoT Networks". In: ***Security and Privacy*** 8.4 (2025), e70070.

P5. Fatemeh Stodt, Christoph Reich, and Fabrice Theoleyre. "Beyond Static Security: A Context-Aware and Real-Time Dynamic Zero Trust Architecture for IIoT Access Control". In: ***IEEE Internet of Things Journal*** (2025).

Journal Papers Under Review

P6. Fatemeh Stodt, Christoph Reich, and Fabrice Théoleyre (2025). "Context-Aware Anomaly Detection by Community Detection in the Internet of Things". In: **Computer Communication**. Under review.

P7. Fatemeh Stodt, Philipp Ruf, Christoph Reich, and Fabrice Théoleyre (2025). "Distributed Zero Trust Architecture Based on Policy Negotiation Secured by DPP in Blockchain". In: **Annals of Telecommunications**. Under review.

Conference Papers

P8. Fatemeh Stodt and Christoph Reich. "A Review on Digital Wallets and Federated Service for Future of Cloud Services Identity Management". In: ***SERVICE COMPUTATION 2023, The Fifteenth International Conference on Advanced Service Computing, June 26-30, Nice, France***. IARIA. 2023 **(Best paper award)**.

P9. Fatemeh Stodt, Fabrice Theoleyre, and Christoph Reich. "Advancing Network Survivability and Reliability: Integrating XAI-Enhanced Autoencoders and LDA for Effective Detection of Unknown Attacks". In: ***2024 20th International Conference on the Design of Reliable Communication Networks (DRCN), May 6-9, Montréal, Canada***. IEEE. 2024, pp. 9–16.

P10. Fatemeh Stodt, Philipp Ruf, and Christoph Reich. "Blockchain-Enabled Digital Product Passports for Enhancing Security and Lifecycle Management in Healthcare Devices". In: ***2024 8th Cyber Security in Networking Conference (CSNet)***. IEEE. 2024, pp. 44–51.

I was taught that the way of progress was neither swift nor easy. (Marie Curie)

Contents

Chapter 1

Introduction

Contents

1.1 Context and Motivation

The Industrial Internet of Things (IIoT) has emerged as an essential force in transforming various sectors, including manufacturing, healthcare, energy, and transportation. The ability to interconnect devices and sensors within industrial processes has enabled unprecedented levels of automation, efficiency, and real-time monitoring. IIoT solutions not only help improve operational efficiency but also support early maintenance to prevent equipment failures, make production flows more efficient, and enhance decision-making capabilities.

Despite these considerable advantages, the rapid expansion of IIoT infrastructures introduces significant security challenges. Historically, industrial networks relied heavily on perimeter-based security mechanisms, presuming a clear separation between trusted internal devices and external threats [9]. However, traditional security boundaries are undermined by the integration of many devices, services, and stakeholders across different domains, resulting in a vast attack surface that is difficult for traditional security methods to properly control [10].

Real-world incidents illustrate the severe consequences of compromised IIoT security. High-profile cyberattacks, such as the Stuxnet worm [11], the Triton malware [12], or recent ransomware targeting industrial control systems, demonstrate how vulnerabilities can lead to disruptions, financial losses, and risks to human safety [13]. These events underscore the urgency of developing new approaches capable of

protecting interconnected, highly distributed, and resource-constrained industrial environments.

Furthermore, IIoT environments typically feature a diverse mix of devices from robust, powerful servers to small, resource limited sensors [14]. Traditional cybersecurity mechanisms often impose significant computational overhead, which is incompatible with lightweight IIoT devices [15]. Thus, securing these environments requires innovative, efficient approaches that can maintain robust security without sacrificing performance or scalability.

To address these inherent vulnerabilities and limitations, modern cybersecurity paradigms are exploring adaptive strategies, such as Zero Trust Architecture (ZTA), and decentralized, tamper-proof trust management systems, including blockchain. While these concepts provide a foundation for improved resilience, many implementations lack context-awareness or remain too centralized to meet the demands of dynamic, distributed industrial environments. While these emerging solutions offer considerable promise, practical deployments face persistent challenges regarding complexity, computational overhead, scalability, and real-time responsiveness, particularly in cross-domain scenarios.

This thesis is motivated by the critical need to bridge these gaps by providing practical, robust, and scalable security solutions tailored explicitly to IIoT ecosystems. Specifically, this work aims to integrate context-awareness and the dynamic, adaptive capabilities of ZTA with the decentralized, secure, and transparent features of blockchain technologies, directly addressing the evolving security needs of interconnected industrial environments.

1.2 The Challenge: Security and Scalability in IIoT

Securing the IIoT introduces unique challenges due to its inherent complexity, heterogeneity, and critical nature. For instance, a compromised IIoT system in a power grid could lead to widespread outages or safety hazards, making security breaches far more consequential than in traditional IT environments. Unlike conventional enterprise networks, IIoT systems must secure interactions across diverse operational domains, each with distinct security and functional requirements. Consider, for example, a smart factory environment that integrates robotic assembly lines, automated inventory management systems, predictive maintenance tools, and external supplier interactions. Each subsystem represents a unique security domain, yet all must collaborate seamlessly while maintaining robust security controls.

Traditional cybersecurity approaches primarily relied on perimeter-based defenses, assuming a well-defined boundary between internal, trusted devices and external threats. However, this assumption is fundamentally invalid in modern industrial contexts, where devices and processes span multiple administrative domains and interconnect across open and potentially insecure networks, including the Internet. This openness significantly expands the attack surface, making perimeter defenses inadequate and obsolete.

Moreover, ensuring security in IIoT environments is complicated by the presence of resource-constrained devices. Industrial sensors and embedded controllers frequently lack the computational power and memory resources necessary to implement conventional cryptographic algorithms or complex authentication mechanisms

[16]. Efficient authentication is essential in industrial environments, where devices must verify their identity without causing latency or overloading limited hardware.

Scalability further exacerbates these security challenges, as increasing the number of integrated devices significantly worsens security issues and quickly overwhelms traditional centralized mechanisms.[17]. Centralized authentication and trust management systems can become performance bottlenecks and introduce single points of failure, severely affecting the resilience and reliability of industrial operations. Centralized security architectures lack the flexibility needed to quickly respond to evolving threats and changing industrial conditions.

An additional layer of complexity is introduced by the necessity of real-time operations within industrial contexts. Delays resulting from security mechanisms, such as extensive cryptographic verification processes or frequent policy checks, could disrupt sensitive processes, potentially resulting in downtime or even safety hazards. Therefore, effective IIoT security must achieve a delicate balance, providing robust protection without compromising real-time responsiveness or operational efficiency.

Addressing these intertwined challenges cross-domain interoperability, resource constraints, scalability, and real-time requirements requires fundamentally rethinking security approaches in industrial environments. Effective solutions must integrate adaptive and context-aware mechanisms capable of dynamically managing security policies, efficiently authenticating diverse devices, and proactively mitigating threats in real-time.

This thesis directly tackles these critical challenges by proposing and evaluating a novel security framework specifically designed for distributed IIoT ecosystems, balancing rigorous security requirements with practical considerations of scalability and operational efficiency.

1.3 Emerging Solutions: Zero Trust and Blockchain

In response to the inherent challenges associated with security and scalability in IIoT, two innovative paradigms have emerged prominently: ZTA and blockchain-based solutions.

ZTA fundamentally shifts traditional security paradigms by adopting the principle of "never trust, always verify" [18, 19]. Unlike conventional perimeter-based approaches, ZTA continuously verifies every device, user, and data flow, regardless of their location within or outside the network perimeter. This approach significantly enhances security by requiring explicit and continuous authentication and authorization based on dynamic contextual information such as identity, location, device health, and behavioral analytics. While ZTA enforces continuous verification, its policies are typically static. Making ZTA dynamic and context-sensitive remains an open challenge particularly in resource-constrained IIoT environments.

However, implementing a pure ZTA solution in industrial environments poses considerable challenges. Real-time adaptability requires comprehensive and continuous context evaluation, introducing potential computational overhead and latency, particularly for resource-constrained industrial devices. Moreover, managing dynamic trust relationships across multiple industrial domains can become complex, demanding efficient mechanisms to ensure responsiveness and scalability without compromising security.

In parallel, blockchain technology has garnered significant attention as a complementary solution for securing distributed industrial environments. The decentralized nature of blockchain provides inherent security benefits, including transparency, immutability, and robust integrity verification. These features make blockchain particularly suitable for scenarios involving multiple stakeholders who require secure collaboration, such as supply chain management, device identity management, and cross-domain authentication. Additionally, blockchain's decentralized consensus mechanisms eliminate the single points of failure associated with centralized trust authorities, thus improving resilience and reducing vulnerability to attacks targeting central control points [20].

Nevertheless, integrating blockchain into IIoT security solutions introduces its own set of challenges. Traditional blockchain implementations often suffer from significant computational and storage overhead, which is problematic for resource-limited industrial devices [21]. The high latency associated with consensus mechanisms and blockchain transactions also poses difficulties in real-time operational contexts. Hence, lightweight blockchain solutions specifically optimized for constrained environments are essential to practically leverage blockchain's advantages within IIoT settings.

Recognizing both the potential and limitations of these emerging paradigms, this thesis proposes a novel integrated approach that enhances the traditionally static and rigid ZTA by introducing dynamic, context-aware capabilities. By combining these adaptations with the decentralized and transparent characteristics of blockchain technology, the proposed solution provides a practical, scalable, and resilient security model tailored to the unique needs and constraints of distributed industrial ecosystems.

1.4 Aims and Objectives

The primary aim of this thesis is to design, develop, and evaluate a security framework specifically customized to address the unique challenges of securing distributed IIoT environments. This is accomplished by integrating adaptive ZTA principles and blockchain-based decentralized trust management to provide robust security, scalability, and real-time responsiveness suitable for industrial applications.

To achieve this overarching goal, the research sets out the following specific objectives:

- Develop and validate a cross-domain authentication mechanism to efficiently manage secure interactions between diverse industrial devices and stakeholders across different administrative domains.
- Create a dynamic and adaptive ZTA framework featuring context-aware security policies that continuously adapt based on real-time contextual information, threat evaluations, and operational dynamics within heterogeneous industrial networks.
- Establish a robust blockchain infrastructure optimized for industrial applications, facilitating decentralized and transparent trust relationships across multiple domains without introducing significant computational overhead.

- Introduce and implement advanced context-aware anomaly detection techniques capable of proactively identifying, classifying, and mitigating emerging cybersecurity threats with high accuracy and minimal false alarms.
- Conduct comprehensive evaluations to demonstrate the practicality, performance, scalability, and adaptability of the proposed integrated ZTA-blockchain security framework through realistic industrial scenarios.

By addressing these objectives, the thesis aims to significantly advance adaptive, decentralized cybersecurity solutions in industrial contexts, contributing theoretical innovations and practical enhancements to the security and resilience of IIoT ecosystems.

1.5 Research Questions

Guided by the aims and objectives, the thesis addresses the following key research questions:

RQ1. What are the challenges in integrating blockchain and Zero Trust principles into a cohesive security framework for IIoT?

RQ2. How can hybrid and context-aware anomaly detection methods improve the real-time identification and assessment of sophisticated security threats within dynamic IIoT networks?

RQ3. How can scalable and secure identity management be achieved in distributed IIoT networks?

RQ4. What are the limitations of traditional ZTA in dynamic IIoT environments, and how can they be adapted for real-time security?

RQ5. How can blockchain address IIoT challenges of scalability, privacy, and tamper-proofing?

RQ6. How can identity management, blockchain, and anomaly detection be integrated into a cohesive Distributed ZTA framework for securing IIoT networks?

Through addressing these research questions, this thesis seeks not only theoretical advancements in industrial cybersecurity but also tangible, practical solutions directly applicable to real-world IIoT environments.

1.6 Contributions of the Thesis

This thesis provides significant advancements in securing distributed IIoT environments by synthesizing the adaptive security principles of ZTA with the decentralized trust management capabilities of blockchain. Specifically, the following contributions are made:

1. **Integrated Identity Management for IIoT:** We first introduce a cross-domain authentication mechanism leveraging digital wallets to securely onboard and manage industrial devices. This mechanism addresses the challenge of dynamically establishing trust relationships across multiple factories or industry verticals.
2. **Context-Aware Anomaly Detection:** Building upon this secure identity foundation, we develop advanced anomaly detection strategies that integrate neural network architectures (autoencoders) with graph-based community detection techniques. These methods dynamically adapt to changing operational contexts, minimizing false positives and robustly detecting previously unknown threats.
3. **Lightweight Blockchain and Digital Product Passports:** To efficiently support these security mechanisms in resource-constrained IIoT environments, we propose a lightweight blockchain framework tailored specifically for industrial applications. Furthermore, digital product passports are integrated to facilitate secure lifecycle management and traceability of industrial assets.
4. **Dynamic Zero Trust Architecture:** Leveraging the established identity management and anomaly detection capabilities, we design and implement an extensible Zero Trust Architecture. This architecture continuously evaluates threat levels using real-time risk assessments, ensuring robust access control and seamless interoperability across heterogeneous industrial domains.
5. **Fully Distributed Access Control Paradigm:** Finally, we extend the developed ZTA framework by introducing a decentralized policy negotiation protocol. This advancement empowers multiple stakeholders to securely collaborate, significantly reducing reliance on centralized authorities and further enhancing resilience and flexibility.

1.7 Structure of the Thesis

The remainder of this thesis is structured as follows:

- **Chapter 2** presents a review of relevant literature, covering IoT/IIoT architectures, identity management, anomaly detection, and blockchain approaches. We highlight how conventional models fall short for large-scale, mission-critical IIoT use cases.
- **Chapter 3** analyzes the security challenges in a typical smart factory environment, detailing the threat vectors and complexity inherent in distributed control systems.
- **Chapter 4** introduces advanced identity management and authentication mechanisms for cross-domain collaboration, focusing on digital wallets and secure bridging of trust across different domains.
- **Chapter 5** elaborates on anomaly detection strategies. We present novel context-aware and hybrid detection algorithms tailored to industrial environments that must handle unknown attacks in real time.

- **Chapter 6** discusses blockchain-based solutions for lightweight and privacy-preserving shopfloor auditing, ensuring high throughput, data integrity, and adaptability to resource constraints in IIoT.
- **Chapter 7** proposes a dynamic Zero Trust Architecture, integrating the previously introduced concepts of identity management, distributed anomaly detection, and blockchain. We also detail a robust risk assessment process.
- **Chapter 8** extends the Zero Trust paradigm to a fully distributed model, implementing policy negotiation and digital product passports for collaborative manufacturing.
- **Chapter 9** concludes the thesis, summarizing our main contributions and outlining perspectives for future research in the domain of secure and scalable IIoT.

Figure 1.1 illustrates the overall structure of the thesis and the interrelationship between its key components. The arrows in the diagram represent the flow of dependency, indicating that each element builds upon or utilizes the outputs of the preceding elements.

The thesis is divided into several core modules, starting with the Network Anomaly Detector and Context Anomaly Detector, which provide foundational insights for Threat Assessment. These components, in turn, contribute to the realization of advanced security and identity mechanisms such as Dynamic ZTA, Lightweight BC, Distributed ZTA, DPP, Cross Authentication, and Wallet – Identity.

The white boxes denote the research topics from which peer-reviewed papers have been published or submitted. These represent the primary academic contributions of the thesis. In contrast, the gray boxes represent additional developed components that support the broader architecture but were not individually turned into standalone publications. This figure thus captures both the structural flow and the academic output of the thesis work.

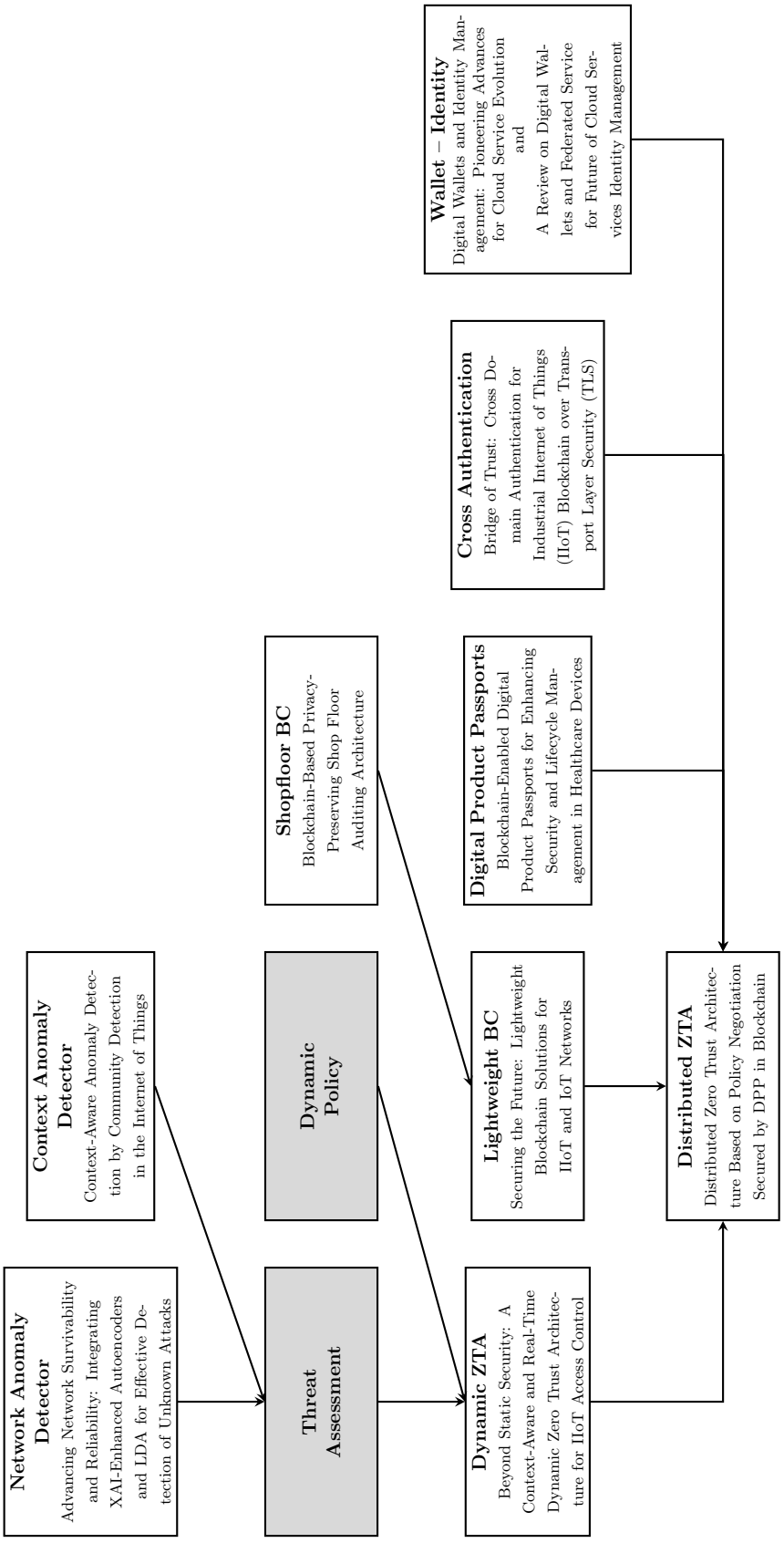


Figure 1.1: Structure of Thesis.

Chapter 2

Foundation & State of the Art

Contents

2.1 Introduction

Our primary objective is to move from a broad overview of Internet of Things (IoT) and its industrial specialization Industrial Internet of Things (IIoT), to a focused examination of critical security foundations such as identity management, access control, anomaly detection, and blockchain-based solutions. In this chapter, we review the state-of-the-art research on security mechanisms and supporting technologies in the context of IIoT.

Chapter Organization: The following sections are structured to progressively build a comprehensive understanding of securing IIoT environments, systematically highlighting critical concepts and challenges:

- **Section 2.2** distinguishes ***IoT*** from ***IIoT***, highlighting the unique security challenges specific to industrial environments. This foundational understanding sets the stage for addressing security requirements tailored to industry.
- Building upon these industrial-specific challenges, **Section 2.3** introduces ***Identity and Access Management (IAM)*** in IIoT, emphasizing the importance of scalable authentication and authorization mechanisms designed explicitly for large-scale, real-time industrial operations.
- With identity management established, **Section 2.4** explores ***Network Anomaly Detection***, detailing how proactive detection mechanisms serve as critical defenses in protecting complex IIoT networks from evolving threats.
- To further enhance security, **Section 2.5** examines the role of ***Industrial Blockchain*** in decentralizing trust and maintaining data integrity. This discussion addresses not only its benefits but also the inherent challenges of blockchain technology within industrial contexts.
- Extending the blockchain discussion, **Section 2.6** presents ***Digital Product Passports (DPP)***, demonstrating how blockchain technology facilitates comprehensive product traceability, thereby supporting sustainable practices and the circular economy in industrial operations.
- Finally, integrating these prior concepts, **Section 2.7** discusses the ***Zero Trust Architecture (ZTA)*** model, presenting it as an overarching modern security framework enforcing continuous verification of devices, users, and data flows, essential for resilient IIoT infrastructures.

2.2 IoT and IIoT: Fundamentals and Challenges

2.2.1 Defining IoT and IIoT

IoT is a network of interconnected devices that communicate and share data via networked infrastructures [22]. The underlying technology involves embedded sensors, microcontrollers, and actuators that enable data collection and remote actuation through various wireless protocols (such as Wi-Fi, Bluetooth, Zigbee, and NB-IoT). In terms of topology, IoT systems typically employ a layered architecture:

- **A device layer** (comprising the sensors and actuators),
- **An edge or gateway layer** (which may perform preliminary data processing or protocol translation), and
- **A cloud layer** for centralized data storage and complex processing.

Additionally, alternative topologies like mesh networks are sometimes used to enhance connectivity and reliability in dynamic environments. Typical applications of IoT include smart homes (for automation, energy management, and security), wearable health devices, connected vehicles, and smart city solutions (such as environmental monitoring and traffic management).

IIoT extends these principles to critical industrial sectors [23], employing industrial-grade sensors, actuators, and control systems designed for harsh environments. It leverages real-time communication protocols and often integrates edge computing to satisfy the low-latency and high-reliability demands of industrial applications. Typical IIoT networks utilize hierarchical, redundant architectures combining centralized control with local processing, ensuring continuous operation even during faults. Common applications include automated manufacturing, predictive maintenance, smart grid management, and process automation tasks prioritizing efficiency, safety, and reliability.

2.2.2 Key Differences and Challenges in IIoT

Although both IoT and IIoT create intelligent networks by leveraging connected devices, IIoT differs significantly in scale, criticality, and security requirements. These differences, however, introduce several challenges that must be addressed for effective and safe deployment.

Mission-Critical Operation: Unlike typical IoT applications – where occasional brief failures may be tolerable – IIoT systems operate in environments such as manufacturing, power grids, and transportation, where even minimal downtime can lead to add costs or safety risks. This mission-critical nature requires near-zero tolerance for failures.

Strict Real-Time Constraints: Industrial processes demand deterministic or low-latency responses to sensor data. To meet these real-time constraints, IIoT architectures often rely on local processing using intelligent gateways or edge servers, thereby avoiding delays associated with remote, centralized data processing.

Complexity in Legacy Integration: Many industrial organizations depend on legacy systems such as Programmable Logic Controllers (PLCs) or Supervisory

Control and Data Acquisition (SCADA) systems, which were not designed with modern security measures in mind [24]. Upgrading or integrating these systems frequently forces production lines to halt, placing stakeholders in a difficult position between maintaining operations and mitigating cyber threats.

Scalability and Heterogeneity: IIoT networks involve thousands of devices with diverse hardware and protocols, creating real-time bottlenecks and demanding scalable, distributed architectures [25]. Moreover, the dynamic nature of industrial environments requires security mechanisms that continuously update access policies and risk assessments to adapt to rapidly changing device states and threat landscapes.

Stringent Security and Safety Requirements: IIoT systems must adhere to strict standards (e.g., IEC 62443 [26]) to ensure data confidentiality, integrity, and availability. This necessitates not only robust risk assessments, redundant fail-safes, and continuous monitoring, but also dynamic security policies – such as automated policy updates and adaptive access controls – that adjust to emerging vulnerabilities and enable proactive anomaly detection and rapid response.

Data Privacy and Ownership: The industrial data, encompassing proprietary manufacturing techniques and sensitive sensor readings, must be rigorously protected. Unauthorized access can lead to intellectual property theft and undermine the operational integrity of critical infrastructures.

2.2.3 Characteristics of IIoT Networks

IIoT networks consist of diverse devices and services covering multiple locations or supply chains [27]. Traditional centralized architectures, which involve a single control unit or data center, simplify management but create a critical single point of failure. To mitigate these risks, many IIoT systems have shifted towards distributed or decentralized architectures. Distributed systems delegate processing and decision-making tasks to multiple nodes, such as intelligent gateways or edge servers. This approach offers several advantages, including fault tolerance, reduced latency, scalability, and redundancy. Distributed architectures facilitate dynamic load balancing and redundancy, which is particularly beneficial for industrial applications that operate around the clock and integrate legacy systems with limited cybersecurity measures. By replicating and decentralizing control functions, distributed architectures enhance overall system resilience and reliability, which is critical in environments where downtime or system failures can lead to significant financial and safety risks [24].

Industrial applications like predictive maintenance, automated quality control, and real-time production monitoring require strict Service-Level Agreements (SLAs) for continuous performance and reliability. These SLAs define key performance metrics like uptime, response times, and operational thresholds. IIoT networks require ongoing monitoring to ensure adherence to SLAs. Engineers use real-time analytics and anomaly detection systems to assess network performance and security, detecting accidental failures and malicious intrusions to maintain high availability and resilience.

2.2.4 Attack Vectors in IIoT Environments

Building on the unique characteristics of IIoT networks discussed in the previous section such as high heterogeneity, real-time requirements, and increased connectivity, it becomes evident that these networks are especially vulnerable to a diverse range of cyber threats. To ensure a secure and resilient industrial environment, it is essential to identify and understand the relevant attack vectors that exploit these characteristics. In this context, several major groups of attack vectors must be taken into account when designing secure IIoT systems. Table 2.1 summarizes these attack groups along with representative examples, based on findings from recent academic research.

Table 2.1: Major Attack Groups in IIoT and Example Attack Types.

Attack Group	Example Attack Types and Description
Social Engineering Attacks	**Phishing:** Deceiving employees or operators into revealing credentials or installing malware [28].
Malware Attacks	**Ransomware:** Infecting systems to encrypt data or disrupt operations; viruses/worms for data exfiltration [29].
Network/Protocol Attacks	**MitM:** Intercepting and modifying data in transit; Replay attacks: re-injecting captured traffic; DoS: exhausting network or computing resources [30].
Physical/Hardware Attacks	**Side-Channel Attacks:** Exploiting physical leakages (e.g., timing, electromagnetic emanations); Device Tampering and Hardware Trojans: altering hardware/-firmware to introduce vulnerabilities [31].
Supply Chain Attacks	**Compromised Components:** Injecting malicious code or backdoors during manufacturing or software updates [30].
Identity/Authentication Attacks	**Credential Abuse:** Using stolen or weak credentials to gain unauthorized access [31].

The design of IIoT network must incorporate robust security measures to reduce risk from attack vectors. Key topics include IAM to ensure that only authorized entities interact with the system. Access Control and Policy Enforcement mechanisms restrict actions to authorized parties. Dynamic and Distributed Policy Enforcement enables real-time policy adaptation. Network Anomaly Detection techniques monitor traffic for suspicious activities. Industrial Blockchain ensures tamper-proof transactions and secure device interactions. The DPP verifies component authenticity. ZTA follows the principle of "never trust, always verify" [18, 19], continuously authenticating every device and user within the network. These measures ensure continuous protection even as network conditions evolve [32].

2.3 Identity and Access Management in IIoT

IAM is a discipline that ensures that only genuine entities can access to network resources. This is essential to keeping the industrial network secure in case of interactions, where disruptions can impact safety and productivity.

2.3.1 What Is an Identity in IIoT?

An identity (ID) is a unique set of attributes that distinguishes one entity from another. In an IIoT environment, a device's identity might contain hardware identifiers (e.g., serial numbers), network addresses (e.g., MAC or IP), cryptographic keys, or manufacturer-issued certificates. These attributes not only identify the device but also play a crucial role in authentication and authorization (Ad). Formally, we can represent an identity for a device d as a tuple:

$$\mathrm{ID}(d) = \Big(U_d, A_d, K_d\Big), \tag{2.1}$$

The structure of $\mathrm{ID}(d)$ includes universally unique attributes U_d, descriptive attributes such as device type and firmware version A_d, and an optional cryptographic key or certificate K_d. While $\mathrm{ID}(d)$ remains constant throughout the device's lifecycle, the access rights and authorization associated with it can change dynamically to meet evolving security needs.

2.3.2 Identity Lifecycle in IIoT

Once assigned, a device's identity progresses through several stages, known as the identity lifecycle [33, 34]. Figure 2.1 depicts three key phases:

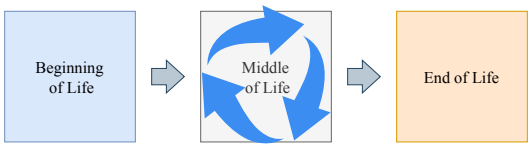


Figure 2.1: Stages of a Device Identity Lifecycle in IIoT.

- **Beginning of Life (BoL):** At this initial phase, the device is built, certified, and registered with management services. During BoL, the device's identity $\mathrm{ID}(d)$ is established, with secure embedding of authentication elements to prevent tampering. This static identity provides a trustworthy basis for subsequent security processes.
- **Middle of Life (MoL):** The device maintains its core identity $\mathrm{ID}(d)$, but dynamic conditions like firmware updates, credential rotations, and continuous monitoring necessitate real-time updates of access control policies linked to $\mathrm{ID}(d)$ be updated in real time. The authorization context adapts based on current risk assessments and operational requirements.
- **End of Life (EoL):** As devices are decommissioned or repurposed, revoking or suspending the associated $\mathrm{ID}(d)$ is crucial to prevent exploitation of stale credentials. This ensures that outdated identities do not grant unauthorized access, thereby maintaining the overall security integrity of the system.

Robust security measures like secure provisioning and real-time monitoring must be synchronized with the IIoT lifecycle stages to avoid preventing minor lapses in tracking or device identity revisions.

Figure 2.2: Evolution of Identity Management Systems [40].

2.3.3 Authentication in IIoT

After defining and assigning identities, authentication verifies that an entity requesting access or data exchange is truly who or what it claims to be. Formally, given an identity $\mathrm{ID}(d)$, authentication involves a procedure $\mathcal{A}$ such that:

$$\mathcal{A} : (d, \mathrm{ID}(d), \mathrm{credentials}) \longmapsto \{\mathrm{true},\ \mathrm{false}\}, \tag{2.2}$$

where $\mathcal{A}$ returns true if the presented credentials match the identity's expected attributes and false otherwise.

Common Authentication Approaches in IIoT devices vary widely in their computational resources and security requirements, leading to the use of multiple methods:

- **Symmetric Cryptography-Based** systems use shared secret keys. Devices generate or store a pre-shared key, which is efficient for resource-constrained endpoints [35].
- **Asymmetric Cryptography-Based** Public-Key Infrastructure (PKI) enhance security by allowing devices to hold private keys and publish public keys in certificates. However, key generation and management require more computational overhead [36]. Advanced variations like post-quantum cryptography are explored for future-proofing industrial networks.
- **Blockchain-Based** decentralized ledgers, like BASA, enable device identity storage and verification without a central certificate authority, but must address scalability and privacy challenges for successful implementation [37].
- **Physical Unclonable Functions (PUF)** are used in industrial settings to generate unique fingerprints from hardware variations, providing a tamper-evident identification method [38].

2.3.4 Identity Management (IdM) in IIoT

Identity Management (IdM) is a key component of an IIoT network which guarantees the authenticity and management of identities over time, scalability, and easy connection with legacy systems. The three main roles in IdM are the Identity Holder (device or user), the Service Provider (application or service holder), and the Identity Provider (authority that issues and vouches for identities) [39]. Depending on how these roles are constructed, IdM models are developed:

1. **Isolated IdMs:** Early systems store credentials locally, with minimal coordination, but this model is not scalable for large IIoT deployments [41].
2. **Centralized IdMs:** A dedicated Identity Provider efficiently validates requests and issues credentials for multiple services, but it also creates a single point of failure [41].

3. **Federated IdMs:** Multiple IdMs establish trust, enabling Single Sign-On (SSO) across organizations, such as in IIoT, where federations link multiple factories or supply chain partners [42].

4. **User-Centric IdMs:** The approach of transferring control to the user or device holder, who can manage attributes to reveal to each service, is suitable for specific IIoT contexts [43].

5. **Decentralized IdMs:** Blockchain-based approaches, such as Self-Sovereign Identity (SSI) frameworks, store and verify identities without a single controlling entity, using Decentralized Identifiers (DIDs) and Verifiable Credentials to prove attributes without a centralized authority [44].

Table 2.2 summarized the pros and cons of these models. The choice of IdM architecture in a IIoT environment can significantly impact scalability, security, and administrative overhead, especially when integrating several industrial domains or legacy systems.

Table 2.2: Comparison of Identity Management Models in IIoT Contexts.

	Isol. IdMS	**Centr. IdMS**	**Fed. IdMS**	**User-C. IdMS**	**Decent. IdMS**
Scalab.	Lim.	High	Med.	Natural	High (DLT)
Sec.	Prov.-Dep.	SPoF	Struct. Impr.	User/Dev. Ctrl.	Dist. Consensus
Data Prot.	Prov.-Ctrl.	Risky Cent.	Multi-Providers	Fine-Gr. Ctrl.	Off-Ch. Privacy
Interop.	None	Low	Cross-Dom.	Higher if Stds.	Net.-Struct.-Dep.
Admin Eff.	Low (Small)	Medium	Shared Orgs	Shift to User/Dev.	Med. (Multi-Inst.)

Abbreviations: Isol. = Isolated, Centr. = Centralized, Fed. = Federated, User-C. = User-Centric, Decent. = Decentralized, Scalab. = Scalability, Sec. = Security, Data Prot. = Data Protection, Interop. = Interoperability, Admin Eff. = Administrative Effort, Prov.-Dep. = Provider-Dependent, SPoF = Single Point of Failure, Dist. = Distributed, Off-Ch. = Off-Chain.

Overall, IIoT security is based on robust identification and authentication mechanisms that are critical for access control, anomaly detection, and data protection. As IIoT settings develop in size and complexity, adopting properIdM techniques becomes more critical for ensuring reliable, high-availability industrial operations.

2.4 Access Control and Policy Enforcement in IIoT

Having established the importance of robust identity management in IIoT ensuring that every device and user is uniquely identified and authenticated, the next critical layer of security is access control and policy enforcement. While identity management verifies "who" is interacting with the system, access control and policies define and regulate "what" these identities are allowed to do once authenticated, and "under what conditions" these actions are permitted.

Access control can be formally defined within the widely adopted Role-Based

Access Control (RBAC) framework. Let:

$$\text{RBAC} = (U, R, P, S, UA, PA), \tag{2.3}$$
$$U = \{u_1, u_2, \ldots, u_n\}, \quad \text{(Set of users)} \tag{2.4}$$
$$R = \{r_1, r_2, \ldots, r_m\}, \quad \text{(Set of roles)} \tag{2.5}$$
$$P = \{p_1, p_2, \ldots, p_k\}, \quad \text{(Set of permissions)} \tag{2.6}$$
$$S = \{s_1, s_2, \ldots, s_l\}, \quad \text{(Set of system resources)} \tag{2.7}$$
$$UA \subseteq U \times R, \quad \text{(User-to-role assignment)} \tag{2.8}$$
$$PA \subseteq R \times P. \quad \text{(Role-to-permission assignment)} \tag{2.9}$$

An access decision is made by evaluating whether a user $u \in U$ has a role $r \in R$ that is assigned the permission $p \in P$ to perform an operation on a resource $s \in S$.

Policies are high-level specifications that capture rules and constraints governing system behavior [45]. In formal terms, a policy can be represented as a predicate over tuples (u, s, a, t), where u is the user, s is the system resource, a is the action, and t is the time of access. For example, a policy may enforce that "only maintenance personnel may initiate system reboot commands", which restricts the action a based on the identity u and context (time t, resource s).

2.4.1 Static Policies and Limitations

During the IIoT setup phase, administrators typically define access rules manually, following traditional rule-based approaches. These static policies, once established, are rarely updated and are not capable of handling real-time environmental changes or unexpected device failures [46]. For instance, a manufacturing plant with a designated set of operators controlling a robotic workstation may need to reconfigure access rights manually, which can delay operations, introduce security gaps, and undermine system efficiency.

2.4.2 Dynamic and Distributed Policy Enforcement in Collaborative IIoT

Dynamic policies, in contrast, are designed to adapt automatically based on contextual information. In these systems, policies are defined using formal rule-based frameworks that integrate real-time data (such as device location, operational state, or threat alerts) into the access control decision process [47]. A typical rule might be specified as follows:

$$\begin{aligned}&(\text{User Role} = \text{'operator'}) \quad \wedge(\text{Time} \in \text{Operational Hours}) \wedge (\text{Device Status} = \text{'secure'})\\ &\Rightarrow \text{allow access.}\end{aligned} \tag{2.10}$$

In this formalism, the policy is a predicate over variables representing the identity of the user, the temporal context, and the security state of the device. The dynamic policy engine continuously monitors these contextual attributes by interfacing with sensors and management systems.

The implementation of dynamic policies generally involves the deployment of distributed components such as Policy Decision Points (PDPs) and Policy Enforcement Pointss (PEPs), particularly at the network edge. These components work in

concert: the PDP evaluates current context against formal policy rules, while the PEP enforces the decisions locally – thereby reducing latency and ensuring that any environmental changes are immediately reflected in access control decisions. Moreover, formal methods such as model checking with temporal logic are employed to verify that these dynamic policies satisfy safety and operational constraints under all conditions, including adversarial scenarios [48].

For example, if a sensor is detected to be running outdated firmware, the dynamic policy may automatically revoke its permission to send data to critical production databases. This decision is made in real time by the PDP after evaluating the device's security posture against the defined policy rules, and it is enforced immediately by the PEP. Such an approach minimizes manual intervention and enhances the resilience of IIoT systems by continuously adapting to both expected and unforeseen changes in the operational environment.

2.4.3 Negotiation Protocols in Collaborative IIoT

To ensure security and privacy when multiple organizations interact, Trust Negotiation Protocols progressively exchange credentials and partial disclosures [49, 50]. For example, if Factory A wishes to share real-time data with Factory B, both parties can negotiate access based on each other's policy constraints. Dynamic trust management might dictate that certain data is only shared if Factory B's equipment meets a safety standard or confidentiality level. Trust Negotiation Protocols control carefully designed interactions to avoid oversharing instead of immediately sharing internal policies or credentials [51].

Karmakar *et al.* [52] presented a policy-driven framework for IoT devices that automatically negotiates and updates trust rules among different administrative domains, demonstrating how dynamic conditions (e.g., shifting production deadlines or workforce changes) can trigger real-time policy revisions.

The immediate collaboration concept enables manufacturers to adjust policies while maintaining operations. This approach balances security requirements with operational needs for seamless inter-factory collaboration. As industrial ecosystems grow, improved policy-based enforcement solutions become critical for ensuring reliable, efficient IIoT.

2.4.4 Access Control Models for IIoT

Policies specify which actions are permitted, but an access control model enforces them at runtime [53, 54]. Three main models are frequently mentioned:

- **RBAC:** This model assigns privileges to roles (e.g., "Supervisor", "MaintenanceBot") rather than individual users or devices [55]. It reduces administration in stable organizations with hierarchical job functions but can lead to role explosion in dynamic IIoT settings.
- **Capability-Based Access Control (CapBAC):** Access rights are encoded in transferable tokens called capabilities [56]. For example, a device holding a maintenance capability might be permitted to update firmware on a cluster of sensors. While CapBAC naturally supports delegation, secure token distribution and revocation can become complex at scale.

- **Attribute-Based Access Control (ABAC):** Policies in ABAC reference attributes of the subject (e.g., device location or user clearance), the resource (e.g., sensor type), and context (e.g., time of day, production phase) [54]. This fine-grained, context-aware approach is well-suited for dynamic policies, but designing and managing attribute sets can be intricate.

2.5 Zero Trust Architecture (ZTA)

While traditional access control mechanisms and static policy enforcement have long served as the backbone of IIoT network security, they often fall short in addressing the evolving landscape of threats and the dynamic nature of industrial environments. This limitation highlights the need for more adaptive and context-aware security models. One such model is the Zero Trust Architecture (ZTA), which redefines how trust and access are managed in modern networks.

The foundational principles of ZTA were first introduced by John Kindervag [57] and further elaborated in NIST Special Publication 800-207 [58]. Evan Gilman's Zero Trust concept emphasizes five key factors: the Internet's inherent insecurity, internal and external security threats, unreliable trust relationships, rigorous authentication and authorization of all participants, and dynamic access control strategies focusing on continuous monitoring and adaptive authorization [59].

The core principles of "never trust, always verify" [18, 19] emphasize strict security measures like micro-segmentation and least privilege access, dividing networks into smaller segments and ensuring minimal access for users and devices to perform their functions [60].

However, static Zero Trust implementations, while crucial in introducing these principles, have several limitations. Static policies struggle with inflexibility and administrative overhead, making them ill-suited for real-time threats and changes in device states [61]. Static, border-based security measures are ineffective against internal threats once attackers bypass the perimeter, as they offer minimal oversight within the network, which poses a critical vulnerability given the increasing mobility of devices and cloud-based operations [62]. This static approach fails to address the need for continuous trust evaluation, which assesses user behavior and dynamically adjusts permissions based on context and risk levels [63].

The implementation of ZTA relies on a set of core logical components that work together to enforce strict access control and continuous verification. As depicted in Figure 2.3, these components [18] include:

- **Policy Engine (PE):** The PE (Policy Engine) is the component responsible for making access decisions based on predefined enterprise policies and rules. It issues allow or deny decisions for each access request, without relying on implicit trust due to network location.

- **Policy Administration (PA):** The PA (Policy Administrator) is responsible for establishing or terminating communication sessions according to the decisions from the Policy Engine. It passes configuration details to the PEPs and may issue session-specific credentials or tokens.

- **PEP:** The PEP (Policy Enforcement Point) is the system component that enforces access decisions from the Policy Engine. It monitors, intercepts, and controls all communications between subjects and resources, enforcing session rules as defined by policy.

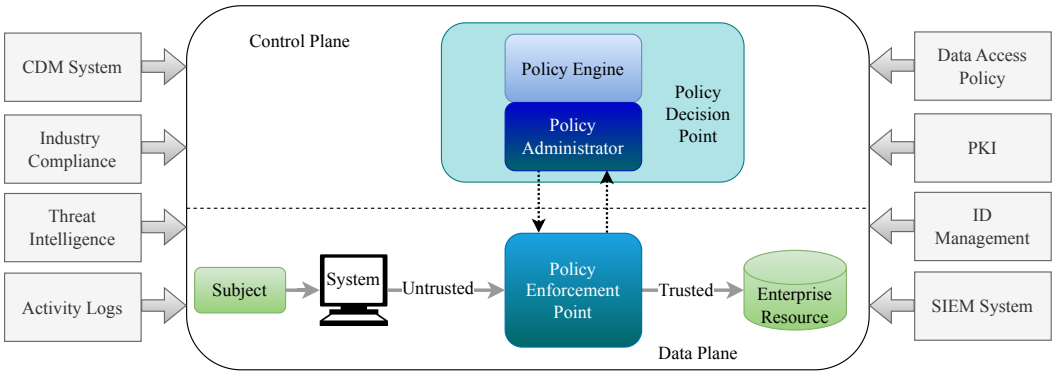


Figure 2.3: Core Zero Trust Logical Components [58].

These components work together to reduce implicit trust and enforce a ZTA, where access decisions are made per request based on predefined security policies, identity verification, and resource sensitivity. Unlike traditional perimeter-based models, ZTA assumes no user or device is inherently trustworthy, regardless of network location.

2.5.1 Context Aware Policies

While static policies remain prevalent in most Zero Trust implementations, they struggle to accommodate the dynamic contexts of user behavior, device posture, and network integrity [47]. Recent research has begun to explore context-aware mechanisms within ZTA, including approaches such as behavioral monitoring, adaptive trust scoring, and runtime policy updates. However, many of these approaches still face challenges in terms of scalability, integration complexity, and responsiveness (particularly in environments with resource-constrained devices or rapidly changing threat conditions).

ZTSFC [64] introduces a dynamic approach where each Resource Access Request is evaluated in real time through a chain of service functions. In this architecture, network traffic is not routed along fixed paths; instead, it is dynamically steered through a series of trusted intermediary nodes that serve as PDPs. These PDPs assess various contextual factors including historical behavior, current network conditions, and predefined risk metrics to deliver precise, context-aware access control decisions. Complementing this traffic-steering paradigm, SYSFLOW [65] adopts a system flow-based model that abstracts all system activities into flows. This model provides a unified framework by separating the data plane from the control plane, enabling developers to specify security intents, enforce micro-segmentation, and manage flow rules both reactively and proactively. While ZTSFC focuses on real-time trust assessments to guide access control, SYSFLOW offers a holistic view of system operations and risk management. Together, these approaches enhance

the adaptability and granularity of ZTA by addressing different layers of security challenges.

Other studies have explored the integration of formal verification methods into dynamic policy enforcement. Niu *et al.* [66] propose a runtime model checking method that converts policy descriptions into logical representations using tools such as TLA+ (formal specification language [67]). This approach involves a multi-stage process policy abstraction, instantiation, pre-check, and post-check to ensure that actual system behavior complies with the specified security policies. However, its dependency on accurately pre-defined rules can limit its robustness in diverse environments. In parallel, Yao *et al.* [68] introduce a Trust-Based Access Control (TBAC) model that adjusts permissions according to the trust level derived primarily from user behavior. Although TBAC enables adaptive permission changes, its reliance on static thresholds and a narrow focus on user context (ignoring factors such as device location or traffic patterns) suggests that it may not fully capture the complexity of dynamic network environments. Together, these approaches illustrate two evolving strategies: one that leverages formal verification to enforce dynamic policies and another that adapts trust levels in real time, each with its own strengths and limitations.

Additional complementary strategies have been proposed to address dynamic security in specific contexts. ZASH [69] targets smart home scenarios using a layered security model that differentiates between user levels, device classes, and permitted actions. By integrating edge computing, ZASH enhances authentication accuracy and maintains local control, thereby reducing performance overhead. Similarly, eZTrust [70] implements a perimeterization approach for microservices in data centers. Utilizing eBPF, eZTrust traces events and attaches context-based tags to packets, which are then verified in real time through dual operational paths-fast for known contexts and slow for unknown ones. Although both ZASH and eZTrust contribute to dynamic security enforcement, they are designed for different environments (smart homes vs. data centers) and rely on an underlying assumption that core infrastructural components remain uncompromised.

2.5.2 ZTA for IIoT

ZTA for Cyber-Physical Systems (ICPS) address challenges such as integrating heterogeneous and legacy devices, ensuring real-time operational responses, and maintaining both cyber and physical safety. Multiple research works have proposed dynamic enforcement mechanisms that evolve from static, identity-centric approaches toward comprehensive, context-aware frameworks.

Feng and Hu [71] propose a Cyber-Physical-ZTA that extends traditional Zero Trust Architectures by incorporating physical context into security decisions. Their approach employs a multi-layer control engine that collects and aggregates operational and security data from various hierarchical levels such as the physical layer (sensors and actuators), the network layer (gateways and routers), and the application layer (software modules and controllers) to evaluate trust scores for each component. These aggregated scores enable the system to generate granular, context-specific access policies in real time.

Complementing this concept, Zanasi *et al.* [72] and Federici *et al.* [73] demon-

strate that dedicated enforcement mechanisms deployed at both network and edge levels can improve the manageability and scalability of industrial control systems. Their work introduces network micro-segmentation, whereby the network is partitioned into small, isolated segments to prevent unauthorized lateral movement. This micro-segmentation effectively limits the impact of any potential breach, working in tandem with multi-layer control engines to ensure that trust is evaluated and enforced at multiple points.

Paul and Rao [74] developed a Zero Trust model for smart manufacturing environments, which includes an identity and access management platform, a Privilege Remote Access jump server, and an Enterprise Device Discovery System. The model uses encryption, policy-based access control, and continuous compliance verification to secure both on-premises and cloud-hosted infrastructures. The model emphasizes micro-segmentation, isolating manufacturing cells and critical infrastructure into distinct segments to limit lateral movement in case of a breach. This dynamic segmentation transforms traditional network divisions into adaptable security zones that enforce granular, context-aware access policies in real time. The model addresses the challenges of managing heterogeneous and legacy industrial systems while ensuring only verified entities gain access. Complementary studies integrate advanced context-aware risk assessment and automated policy adjustment mechanisms.

Xiao et al. conducted a systematic review of context-aware and risk-based access control models within Zero Trust systems, particularly in IoT and IIoT environments [75]. Their work surveys existing mechanisms that leverage real-time contextual information (such as user identity, device posture, and behavioral risk) to dynamically adapt access policies. The paper analyzes models based on ABAC, RBAC, and hybrid schemes, as well as trust assessment techniques like fuzzy logic. While comprehensive, the authors do not propose a concrete implementation, instead highlighting unresolved issues such as reliance on accurate context data, integration complexity, and limitations in centralized control mechanisms.

2.5.3 Limitations of Static ZTA

Traditional Zero Trust implementations often rely on static, predefined policies, which lack the flexibility needed to address rapidly evolving threats in dynamic and distributed environments [47]. This rigidity is particularly problematic in cloud and edge computing, where user behavior, device state, and resource allocation can shift in real time [63, 76]. As a result, static ZTA models struggle to respond effectively, prompting growing support for adaptive approaches better suited to modern computing networks.

The maintenance of static ZTA policies imposes a significant administrative burden, requiring manual updates to accommodate contextual changes. In dynamic cloud environments, where services frequently change, maintaining static access control lists becomes increasingly challenging [63]. Automated policy enforcement significantly reduces the manual workload associated with static policies. The reliance on manual updates strains administrative resources and increases the risk of misconfigurations and security gaps [77].

Static ZTA implementations often employ RBAC, which assigns access based

on predefined roles. While RBAC simplifies access control, it lacks the granularity needed in dynamic environments. Cloud and edge computing require access decisions based on real-time attributes such as device health, location, and behavioral patterns [78]. Traditional RBAC policies are too coarse for modern cloud security, as they either over-provision or under-provision access. Alternative models such as ABAC and continuous risk assessment integrate real-time context into access decisions [79].

Static ZTA models authenticate and authorize users or devices only at the point of access, often without subsequent re-evaluation. This approach creates security vulnerabilities, as once an entity is authenticated, subsequent activities may not be scrutinized. Attackers exploiting compromised sessions or privilege escalation can bypass security measures if trust is not continuously reassessed [80]. Zero Trust requires continuous authentication and re-validation of access rights to detect anomalies in real time. Additionally, static trust parameters, such as fixed device IDs or IP addresses, are susceptible to spoofing [81].

Modern computing environments demand a more flexible security approach than static ZTA provides. Cloud and edge computing introduce dynamic scaling, where workloads and devices frequently change. Static policies, originally designed for stable on-premises architectures, struggle to accommodate such fluidity [82].

2.6 Network Anomaly Detection in IIoT

While robust IAM along with access control system are fundamental to securing IIoT systems, these measures alone cannot eliminate all security risks. Even with strict access control and well-defined policies, adversaries both internal and external can exploit zero-day vulnerabilities, insider threats, or subtle misconfigurations to bypass conventional defenses [83]. Thus, to complement these static and dynamic controls, it is imperative to deploy anomaly detection systems that continuously monitor network behavior for irregularities. This section categorizes the existing anomaly detection methods into five distinct approaches: knowledge-based, statistical-based, machine learning-based, deep learning-based, and graph neural networks (GNN)-based techniques.

2.6.1 Knowledge-Based Techniques

Knowledge-based intrusion detection techniques use predefined rules or patterns of known benign/malicious behavior to identify anomalies. These techniques include heuristic analysis, signature matching, and payload statistical analysis. Heuristic analysis uses expert-defined rules or thresholds to flag abnormal activity, such as limits on command rates or sensor values [84]. Signature matching uses a database of known attack signatures to trigger an alarm on matches, often yielding high accuracy and few false alarms for previously seen attacks. However, in IIoT/OT networks with proprietary or real-time protocols and diverse devices, purely signature-based detection faces challenges [85]. Payload statistical analysis builds profiles of normal packet payload content and detects anomalies as deviations from the learned profile. Recent research has applied this idea to IIoT traffic, such as Zhou *et al.* proposing a payload-based anomaly detector for industrial networks using an autoencoder GAN

model to learn normal data patterns. This statistical payload inspection can catch zero-days injecting unusual data without requiring pre-defined attack signatures [86].

2.6.2 Statistical-Based Anomaly Detection

Statistical-Based Anomaly Detection is established by analyzing traffic data's statistical properties, such as means, variances, and distribution shapes. Gaussian models represent the normal distribution of network metrics, such as packet arrival times and flow sizes, under typical operating conditions. Deviations are quantified using metrics like z-score [87]. Histogram-based methods capture the frequency distribution of observed events and can flag statistically significant differences as anomalies [88]. However, these methods have limitations in dynamic IIoT environments due to the assumption of stationarity and linear models. Fluctuations in traffic patterns due to production loads or time-of-day effects can cause baseline shifts, leading to false alarms. Recent research has incorporated non-linear and hybrid modeling techniques to overcome these challenges. Hybrid approaches integrate Gaussian mixture models with clustering algorithms to adaptively update the baseline as traffic patterns evolve, reducing false positives [89]. Non-linear statistical methods can capture intricate interdependencies between network features, improving the detection of sophisticated anomalies in IIoT scenarios [90].

2.6.3 Machine Learning-Based Anomaly Detection

This method learns patterns directly from data and requires labeled datasets to train models that classify normal and anomalous behavior effectively. For instance, the Support Vector Machine technique can be used for classification [91]. It maps data into a high-dimensional space and identifies a separating hyperplane that distinguishes normal data from potential anomalies. Fosic *et al.* [92] have assessed the detection accuracy of various algorithms (Stochastic Gradient Descent (SGD), Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Gaussian Naive Bayes (GNB), Decision Tree (DT), Random Forest (RF), AdaBoost (AB)) with the UNSW-NB15 dataset. However, these methods rely on labeled datasets and struggle to detect zero-day attacks previously unknown security vulnerabilities that attackers exploit before they are identified or patched. In this context, Graph2vec+RF [53] introduces a unique approach by constructing flow graphs from initial bidirectional network packets. The system embeds these graphs and classifies them with a Random Forest model. This lightweight method excels in early detection with minimal labeled data, but may struggle to adapt to evolving attack patterns.

2.6.4 Deep Learning-Based Anomaly Detection

To detect sophisticated attacks, Deep Learning techniques may be particularly useful. In particular, Variational Autoencoders (VAEs) can detect anomalies [93]. VAEs project input data into a lower-dimensional space. By subsequently minimizing the reconstruction error, VAEs allow for selecting the most representative projection of the original data. A significant reconstruction error is therefore assumed to correspond to abnormal traffic. This method is particularly attractive

since it does not require labeled data: the VAE is trained uniquely with normal traffic, and can thus detect zero-day attacks.

Min *et al.* [94] reduce the memory consumption with a Memory-Augmented Deep Auto-Encoder (MemAE). They implement memory networks to store and recall normal data patterns, improving the accuracy of detecting deviations in network traffic.

Packet transmissions can be represented as time series, with anomaly detection focusing on identifying deviations within these series. Thus, Ullah *et al.* [95] model each network flow as an individual time series using a Recurrent Neural Networ (RNN). They compare Long Short-Term Memory (LSTM), BiLSTM, and Gated Recurrent Unit (GRU) techniques to construct the RNN for flow-based anomaly detection. Trinh *et al.* [96] train a stacked LSTM to model traffic in a cellular network. The system can, for instance, detect a rapid growth of the number of users, representing a DoS attack. However, the approach focuses uniquely on radio access, with a reduced data dimensionality. Elsayed *et al.* [97] propose to use first a LSTM autoencoder to construct a temporal representation of the time-series. Then, a one-class SVM detects the boundaries of the normal data.

Hybrid approaches combine multiple detection methods, leveraging their strengths. Tang *et al.* [98] combine a stacked autoencoder with One-Class SVM: the autoencoder reduces the data dimensionality before classification. Sahu *et al.* [99] use first a Convolutional Neural Network (CNN) to extract features of interest locally, and then use a collection of LSTM to classify the different behaviors. However, combining different techniques increases the complexity and requires more computational resources. The complexity of deep learning techniques remains high. Thus, to reduce the data dimensionality, 3D-Intrusion Detection System (IDS) [100] integrates feature disentanglement with dynamic graph diffusion to separate attack-specific features from non-attack-specific ones. LUCID (Lightweight Deep Learning DDoS Detection) [101] reduces the number of layers and parameters to reduce the computational complexity of a CNN.

Deep learning techniques have significantly influenced anomaly detection, with autoencoders being particularly effective due to their ability to model complex nonlinear relationships within data. Autoencoders reconstruct input into a compressed representation and decode it back to its original form, minimizing information loss. The reconstruction error, the discrepancy between the original input and reconstruction, is a crucial metric in anomaly detection. Samples deviating significantly from learned patterns, as indicated by a high reconstruction error, are identified as anomalies [102].

Wang *et al.* developed a hybrid approach for network anomaly detection by integrating the BIRCH clustering algorithm with autoencoders. Their study showed improved computational efficiency and enhanced detection accuracy on four different network security datasets. However, they acknowledged the need for further refinement and the challenges of limited datasets [103].

Min *et al.* have developed a new network intrusion detection method using a Memory-Augmented Deep Autoencoder (MemAE). This method addresses the overgeneralization issue of traditional autoencoders by learning normal input patterns and reconstructing abnormal samples close to normal ones [94]. The approach's efficacy is demonstrated on the CICIDS2017 dataset, offering a new solution for

handling high data dimensionality in cybersecurity.

Also, Yang *et al.* proposed a network intrusion detection system for Software-defined Networks, utilizing unsupervised machine learning for the real-time detection of both known and zero-day attacks. Griffin's used Kitsune dataset [104] to train and operate a set of autoencoders, achieving high accuracy with reduced complexity and latency [105].

2.6.5 Anomaly Detection With GNN

Graphs are widely used to model communication networks [106]. Thus, it seems natural to use the recently proposed Graph Neural Network (GNN) [107] technique, that adapted Deep Learning to graph structures. GNNs rely on a message-passing approach: the node embeddings are passed from one node to another through the edges. The application of GNNs in anomaly detection involves analyzing network transactions, social networks, or any system that can be represented as a graph to identify irregular and potentially malicious activities. GNN can also be applied inversely to generate attacks more complex to detect [108]. GNNs have been employed for anomaly detection in IIoT systems across a range of applications, including smart transportation, smart energy, and smart factories [109]. Sec2graph [110] tries to detect anomaly when constructing a graph structure of events from log files.

E-GraphSAGE [111] made a pioneering piece of work to adapt GNN for the detection of IoT attacks. While classical GNNs are used for node's classification, E-GraphSAGE detects anomalous flows (edges). Thus, the authors modify the aggregation function of the GNN: the node embedding is computed with the features of all the edges with neighbors. By repeating the aggregation, E-GraphSAGE can capture the dependencies farther away in the network. E-GRACL [112] extends the GNN approach by improving the sampling strategy to enhance the feature representation.

Altaf *et al.* [113] extend this approach to handle multigraph structures, where multiple edges may exist between two IoT nodes due to distinct communication flows. They propose a dual Graph Convolutional Network architecture, incorporating an attention mechanism to capture flow-specific dependencies between edges, thereby improving traffic classification accuracy. Friji *et al.* [114] adopt a different approach, where a node represents a flow. An edge exists between two flows if they share the same source or destination. The weight of an edge is the similarity between the two corresponding flows. Thus, the GNN can be directly applied to the flows, using the graph structure. An attention-based feature extractor in parallel with a spatial feature extractor (*i.e.*, a GNN) identifies anomalies.

Park *et al.* [115] integrate multiple graph types and refine embeddings through attention mechanisms. Capturing the dynamics in the graph may be relevant to detect anomalies. AnomRank [116] monitors abrupt changes in node importance scores by counting the number of edge insertions or deletions associated with the node's significance in a social network.

Temporal Graph Networks (TGNs) [117] manage evolving graph structures, applied to social networks. The node embedding evolves, and an aggregator computes average embedding to reduce the complexity. EULER [118] combines GNN and RNN in a massively parallel architecture to make the detection faster. However, it

still requires a huge amount of computational resources.

2.6.6 Context-Aware Anomaly Detection

Context-awareness is the capacity to improve actions or decision-making within a particular context by leveraging pertinent environmental and situational information, such as time, location, and user activity. It helps distinguish unusual but safe activities from real dangers. SSDCM [119] learns the representation of a multi-layer network, *i.e.*, each layer representing a different context. This technique relies on cluster and node embeddings to derive clusters in complex graph topologies, aggregating the different contexts.

We can also consider uncertainability in context awareness [120]. We should assess the quality of the context. Contextual belief correction (CBC) helps to capture such uncertainty at the cost of a more complex model. Rullo *et al.* [121] also considers the risk and exploits a taxonomy of attacks according to the manufacturer. However, the framework remains conceptual and does not detect zero-day attacks.

Additional information can be used directly from the IoT applications to identify a context. IoT-CAD [122] exploits the sensor's values of each device. Their approach clusters devices with similar values to generate fingerprints of sensors. Then, they use a LSTM neural network and a Gaussian estimator to detect unexpected behaviors. In automotive scenarios, information on the speed may help to detect unexpected messages (*e.g.*, open the door while the vehicle is moving) [123]. DyEdgeGAT [124] exploits directly the signal of sensors to detect anomalies. However, these approaches depend strongly on the application and should be implemented at a higher (application) level.

Table 2.3: Comparison of Different Anomaly Detection Approaches.

Paper	IoT Network	Context-Aware	Graph-Based	Real-Time
[111]	✓	✗	✓	✓
[114]	✓	✗	✓	✓
[124]	✓	✓	✓	✗
[122]	✓	✓	✗	✓
[120]	✗	✓	✗	✗
[100]	✓	✗	✓	✓
Ours	✓	✓	✓	✓

2.7 Industrial Blockchain for IIoT

While network anomaly detection techniques are essential for identifying deviations from normal behavior and detecting malicious activities in IIoT networks, they primarily focus on the reactive identification of irregularities. However, ensuring the integrity and trustworthiness of data exchanged among distributed devices is equally critical in industrial settings [125]. In dynamic IIoT environments, even if an anomaly is detected, it is vital to guarantee that the underlying data remains unaltered and that its provenance can be verified.

To address this need for additional data integrity and non-repudiation guarantees, researchers have explored the use of blockchain frameworks [126]. Industrial

blockchain leverages a tamper-evident ledger and a decentralized trust model to secure data exchanges, ensuring that every transaction is cryptographically verifiable and immutable.

A blockchain is a decentralized, tamper-evident ledger in which data is stored in blocks that are cryptographically linked to form a continuous chain [127]. Let B_i denote the i-th block in the chain. Each block B_i contains a set of transactions and includes the cryptographic hash $H(B_{i-1})$ of the previous block, ensuring that any alteration in an earlier block invalidates the entire chain:

$$B_1 \rightarrow B_2 \rightarrow B_3 \rightarrow \cdots \rightarrow B_n. \tag{2.11}$$

The use of secure hash functions (e.g., SHA-256) guarantees collision resistance and immutability, as even a minor modification in a block's data results in a vastly different hash value [128].

In public blockchains such as Bitcoin or Ethereum, the network is open to anyone who wishes to participate as a validator. These systems typically employ consensus algorithms like Proof of Work (PoW)), in which validators (miners) must solve computationally intensive puzzles to propose new blocks. This process not only secures the network but also makes any attempt to alter previous transactions prohibitively expensive [129].

Conversely, industrial or permissioned blockchains limit validation rights to a set of trusted entities, such as manufacturing partners, thereby reducing the computational overhead [130]. These systems often employ more efficient consensus protocols like Practical Byzantine Fault Tolerance (PBFT) or Raft. PBFT, for example, reaches consensus by allowing nodes to exchange messages in multiple rounds to agree on the validity of a new block, tolerating a predefined number of malicious nodes [131]. Raft, on the other hand, simplifies consensus by designating a leader to coordinate the replication of log entries among nodes [132]. Such protocols offer lower latency and higher throughput, which are essential for the real-time demands of IIoT deployments.

In IIoT environments, where multiple stakeholders and heterogeneous devices interact, the decentralized and immutable properties of blockchain mitigate the risks associated with a single point of trust [133]. By combining cryptographic security with distributed consensus, blockchain frameworks provide robust data integrity, traceability, and resilience against tampering, thereby enhancing overall system reliability.

Integrating a blockchain layer into an IIoT architecture allows nodes to maintain a single source of truth, enforce device identity and traceability, and facilitate secure multi-party collaboration. This is achieved by verifiable transactions or sensor updates, reducing data manipulation, and enabling shared ledgers for factories, suppliers, and regulatory bodies to exchange critical process information without relying on a centralized intermediary [37].

2.7.1 Properties of a Blockchain-based IIoT System

An industrial blockchain can be conceived as an additional layer within the IIoT stack, dedicated to ensuring data consistency, integrity, and auditable operations [134]. Core properties relevant to IIoT include:

- **Immutability and Tamper-Evidence:** The commit of a transaction to a block and appended to the chain makes modifying that record computationally infeasible, preventing malicious actors or insider threats from silently altering production logs or sensor data [135].
- **Decentralized Trust:** Consensus protocols distribute trust across authorized nodes, enhancing trust in multi-factory collaborations where no single entity is universally trusted [133, 136].
- **Auditability and Traceability:** The ledger records every action chronologically, allowing auditors or regulators to verify the chain of events without relying on internal, potentially biased logs.
- **Smart Contracts:** Blockchain platforms enable smart contracts, autonomous scripts that enforce custom rules, eliminating manual oversight and enabling event-driven automation in IIoT processes, such as payment release upon product delivery [137]. However, executing smart contracts, particularly on public blockchains like Ethereum, can incur high transaction (gas) fees due to the computational cost of processing these scripts, which may limit their practical use in cost-sensitive IIoT applications [138].

2.7.2 Blockchain for Manufacturing and Industry 4.0

This section summarizes research papers on the impact of blockchain technology on manufacturing and industrial operations.

Barenji *et al.* [133] develop a blockchain-enabled, multi-agent architecture to support resource sharing and intelligent decision-making in smart factories. They employ a decentralized network that allows devices to collaborate autonomously, ensuring secure data exchange and reliable service provision. Liu *et al.* [139] tackle bandwidth and latency issues in digital twin manufacturing by introducing a peer-to-peer data exchange mechanism underpinned by blockchain, reducing reliance on centralized cloud services while safeguarding security through distributed ledgers.

To improve blockchain performance in industrial networks, Kobzan *et al.* [140] investigated its application through network simulation. Their work highlighted blockchain's ability to address heterogeneity and scalability challenges by evaluating transaction throughput, bandwidth utilization, and deterministic timing. Deterministic timing, which ensures transactions are processed within a predictable time frame, is critical for industrial processes where delays can disrupt operations. While effective in demonstrating blockchain's adaptability to industrial requirements, their simulation-based approach lacks a focus on real-world implementation and lightweight scalability.

For secure data and knowledge management, Zhang *et al.* [136] highlight blockchain's potential in smart factories, underscoring its capacity for real-time communication, adaptive decision-making, and secure data collection. They emphasize the need for refined production scheduling, supply chain coordination, and quality control using a transparent, tamper-proof platform. In parallel, Schmid *et al.* [141] propose a blockchain framework for digital twin manufacturing systems that incorporates peer-to-peer data exchange and manufacturing edge pools, thereby improving data synchronization and reducing network congestion.

Overall, these studies demonstrate blockchain's transformational impact in industrial and manufacturing processes by improving security, scalability, and reliability, while also highlighting the importance of ongoing efforts to address interoperability, infrastructure optimization, and solution scalability.

2.7.3 Lightweight Blockchain

Efficiency within resource-constrained environments has become increasingly critical in the rapidly evolving landscape of blockchain technology [142]. The lightweight blockchain is characterized by its efficient consensus algorithms, reduced on-chain storage through external storage solutions, and optimized cryptographic implementations features that collectively minimize computational and network overhead while ensuring low-energy consumption and high throughput in resource-constrained IoT environments [143]. This model's design revolves around five crucial attributes that distinguish it from conventional blockchain models. These attributes include i) low computational burden, ii) minimal network overhead, iii) reduced storage requirements, iv) enhanced throughput, and v) superior energy efficiency. In this section, we will begin by discussing the constraints that need to be considered. Following that, we will explore potential solutions to overcome these limitations.

Table 2.4: Comparison of Papers on Industrial Blockchain.

Study	Low Comp.	Min. Net.	Red. Storage	High Tput	Energy Eff.	Sec.& Priv.
[144]	✗	✗	✗	✗	✓	✓
[145]	✗	✗	✗	✗	✓	✗
[146]	✗	✗	✓	✗	✓	✗
[147]	✓	✓	✗	✓	✓	✓
[148]	✓	✗	✗	✓	✓	✓
[149]	✓	✗	✓	✗	✓	✓
[150]	✗	✓	✗	✗	✓	✗
[151]	✗	✗	✓	✗	✓	✓
[152]	✓	✗	✓	✗	✓	✓
[153]	✗	✓	✗	✓	✓	✓
[154]	✓	✓	✗	✓	✓	✗
[155]	✓	✓	✓	✓	✓	✗
[156]	✓	✓	✗	✓	✓	✗
Our	✓	✓	✓	✓	✓	✓

Abbreviations: Low Comp. = Low Computational Burden Min. Net. = Minimal Network Overhead Red. Storage = Reduced Storage Requirements High Tput = High Throughput nergy Eff. = Superior Energy Efficiency Sec.&Priv. = Security & Privacy Preserving

2.7.3.1 Limitations of Lightweight Blockchains

Kempa *et al.* propose "collapsing" as a technique to reduce computational power required for cryptographic hashing and consensus algorithms in conventional blockchain systems [157]. The collapsing method aggregates multiple cryptographic hash operations by combining block headers into a single composite hash, reducing the number of individual hash computations needed for consensus and block validation. It maintains cryptographic security through mathematical properties like collision resistance and pre-image resistance, reducing energy consumption and accelerating transaction verification, making blockchain systems suitable for resource-constrained environments like IIoT deployments. In addition, blockchain systems often experience significant network traffic as every node is required to download

and verify all transactions. Network congestion and inefficiency can be a common issue, especially in IIoT environment [150]. In addition, blockchains pose a challenge for devices with limited storage capabilities as they require nodes to store the entire ledger, which can grow significantly over time [151].

Transaction throughput is crucial for real-time applications. Unfortunately, traditional blockchains often experience low throughput because of the time required for consensus and transaction validation [147]. It is worth noting that the energy consumption of blockchain operations, particularly those utilizing PoW, is significant and not ideal for low-power devices. This highlights the importance of finding more energy-efficient solutions.

2.7.3.2 Addressing Constraints With Effective Solutions

Research on reducing the computational demands of blockchain systems has been limited, but several innovative approaches have emerged that target both the computational and communication overhead in resource-constrained environments. For instance, Chaudhry *et al.* [158] focus on developing more efficient consensus algorithms by redesigning the message exchange and voting processes. Their work introduces variants of classical consensus protocols that reduce the number of redundant validation steps and lower the overall communication complexity. By optimizing these procedures, their approach achieves faster convergence and less computational overhead, thereby laying a foundation for advanced, real-time blockchain solutions in IIoT environments.

Bandara *et al.* [148] take a different angle by introducing Tikiri – a blockchain platform tailored for IoT devices. Tikiri employs Apache Kafka as its underlying consensus mechanism, where Kafka's high-throughput, low-latency distributed messaging system efficiently orders transactions. Additionally, Tikiri integrates functional programming paradigms with actor-based smart contracts to enable concurrent and lightweight execution of contract logic. While this innovative integration improves throughput and reduces latency, it also introduces potential scalability challenges, particularly in scenarios where Kafka may become a bottleneck under extreme load.

Dai *et al.* [153] propose the GradedDAG protocol to enhance the efficiency of Byzantine Fault Tolerance (BFT) Directed Acyclic Graph (DAG) systems. Their approach incorporates Reliable Broadcast (RBC) and Consistent Broadcast (CBC) mechanisms to ensure that all honest nodes receive a consistent set of messages, thereby reinforcing consensus without the heavy computational cost typical of traditional BFT schemes. Although GradedDAG markedly improves efficiency, its inherent complexity could limit its applicability in environments with severely constrained resources.

Khan *et al.* [147] present AEchain, which employs a Proof of Authority (PoA) consensus algorithm alongside lightweight authenticated encryption. By replacing energy-intensive mining with a PoA-based approach, AEchain reduces the computational load while ensuring that transactions are cryptographically verified. However, the reliance on a trusted set of validators and the intricacies of establishing and managing these authorities introduce additional complexity during deployment.

Dai *et al.* [154] introduce LightDAG, a protocol designed to minimize network overhead by streamlining the consensus process using Plain Broadcast and Con-

sistent Broadcast mechanisms. This simplification results in reduced latency and lower energy consumption, though it might compromise network robustness when facing malicious activities, as the reduced validation rigor can potentially expose the system to certain adversarial strategies.

Arun *et al.* [155] developed Shoal++, a protocol that balances high throughput with reduced storage requirements by employing staggered DAG instantiation and dynamic anchor scheduling. This dynamic architecture adjusts the frequency of ledger updates based on current network conditions, optimizing resource usage. However, the adaptive nature of Shoal++ poses challenges in maintaining consistency across heterogeneous networks, where device capabilities can vary widely.

Spiegelman *et al.* [156] propose BullShark, an approach that optimizes performance in BFT-DAG systems by reducing the number of consensus rounds. While BullShark achieves significant computational savings, its heavy reliance on the integrity of the DAG structure raises safety concerns if the DAG is disrupted, the entire consensus process may be compromised.

On the storage front, Alkhazaali *et al.* [150] integrate blockchain with a lightweight fog computing solution to offload storage and computation tasks to nearby fog nodes. This hybrid approach effectively alleviates the burden on IoT devices, although it introduces new challenges related to scalability and network management. Similarly, Wang *et al.* [151] address storage reduction via proxy re-encryption, allowing sensitive data to be stored securely in the cloud while only a hash is kept on-chain. This technique reduces on-chain storage but requires robust key management strategies to ensure data availability and integrity.

Additional models, such as the Smart Food Chain (SFC) [149] and LightChain [146], propose sub-blockchain architectures and minimized on-chain data approaches to further lower storage requirements. While these solutions improve throughput and reduce storage needs, their tailored design often limits their broader applicability across different industrial domains. In parallel, the blockchain architectures presented in [152] leverage advanced cryptographic primitives (e.g., Fabrik SDK and Schnorr signatures) to further enhance security and efficiency, though at the cost of increased deployment complexity.

Finally, frameworks targeting energy efficiency are also emerging. Kably *et al.* [144] propose a cloud-based framework that offloads heavy computational tasks to cloud servers, thereby reducing on-device energy consumption. Ekanayake *et al.* [145] introduce the Proof of Equivalent Work (PeW) consensus mechanism as an energy-efficient alternative to conventional PoW, promoting sustainability in industrial applications by lowering the overall energy demand.

2.8 Digital Product Passport (DPP)

A DPP is a system designed to collect and store information throughout a product's lifecycle to support the circular economy [159]. The main goal is to promote *R-strategies* (reuse, repair, refurbish, remanufacture, recycle) by maintaining traceability from production to end-of-life [160]. Unlike traditional product labeling, DPPs ensure a continuous and collaborative digital record rather than isolated lifecycle snapshots [161].

DPPs document product composition, manufacturing details, chemical properties, energy usage, CO2 emissions, and end-of-life handling instructions [159]. These passports must be compatible with multi-tenant tracking systems, ensuring interoperability across stakeholders. However, an EU-wide standard is still pending [161]. The requirements for DPPs include legal, functional, security, accessibility, and modifiability aspects [162].

Donetskaya *et al.* made efforts to elucidate the requirements for DPPs, concentrating on defining the different stages of the life cycle, operational procedures, design choices integral to DPP frameworks, and their possible applications [163]. Standard practices, such as investigating co-contractor components and analyzing previously developed components, are supplemented by evaluating the product's replaceability in terms of materials and components when designing data management within a DPP system.

In general, a DPP serves as a unique document containing life cycle data such as product composition, manufacturing processes, materials, physical and chemical properties, state of charge, substances of concern, usage data like repairs or replaced components, and instructions on how to handle product components at their EoL [159]. The identification of various requirement categories from DPP-enabling systems, including considerations regarding legal aspects, functionality, security, interoperability, modifiability, accessibility, availability, and portability, underscores the multifaceted nature of these systems [162].

Table 2.5: Comparative Analysis of Studies on Digital Product Passports.

Study	BC	Area of DPP	Priv.	Sec.	Scal.	Decent.
[164]	✓	Various Sectors	N.S	✓	✓	✓
[161]	N.S	Sustainability	N.S	N.S	N.S	N.S
[165]	✓	Textile Industry	N.S	✓	N.S	✓
[166]	✓	Recycling	N.S	✓	✓	✓
[167]	N.S	Circular Economy	N.S	N.S	N.S	N.S
[8]	✓	Industry 4.0	✓	✓	✓	✓

Abbreviations: BC = Blockchain, DPP = Digital Product Passport, N.S= Not Specified, Priv. = Privacy Preserving, Sec. = Secure, Scal. = Scalable, Decent. = Decentralized.

2.8.1 Value of DPPs & Integration With Blockchain

DPPs are a great innovation because they provide a centralized repository of information for transparency and traceability. However, integrating blockchain technology can significantly enhance their immutability and tamper resistance [166]. Blockchain ensures that once product data is recorded, it remains unchangeable, preserving integrity over time.

Existing research on DPPs explores various implementations but reveals persistent challenges in scalability and privacy protection. Falco *et al.* present a DLT-based prototype designed to improve circular economy traceability, though they acknowledge significant technical complexities and the absence of privacy mechanisms [166]. Saleheen *et al.* investigate blockchain-integrated DPPs in the textile industry, emphasizing enhanced transparency in supply chains but failing to introduce concrete privacy solutions [165]. Voulgaridis *et al.* propose integrating DPPs

with IoT technology to facilitate sustainability tracking; however, their framework does not incorporate privacy considerations, which remain a key limitation [167].

Despite the advantages of DPPs, privacy remains a significant challenge. Current approaches often neglect privacy-preserving mechanisms, leaving sensitive product lifecycle data vulnerable to exposure. Ensuring compliance with privacy regulations and protecting proprietary product information is crucial for widespread adoption. Table 2.5 illustrates that most studies struggle with privacy concerns.

2.8.2 Use Case: Privacy-Preserving DPPs

Given these limitations, a privacy-focused DPP implementation can enhance security while maintaining data transparency and compliance. The integration of privacy-preserving techniques in DPPs ensures that sensitive product lifecycle data remains protected [8]. Figure 2.4 illustrates this approach.

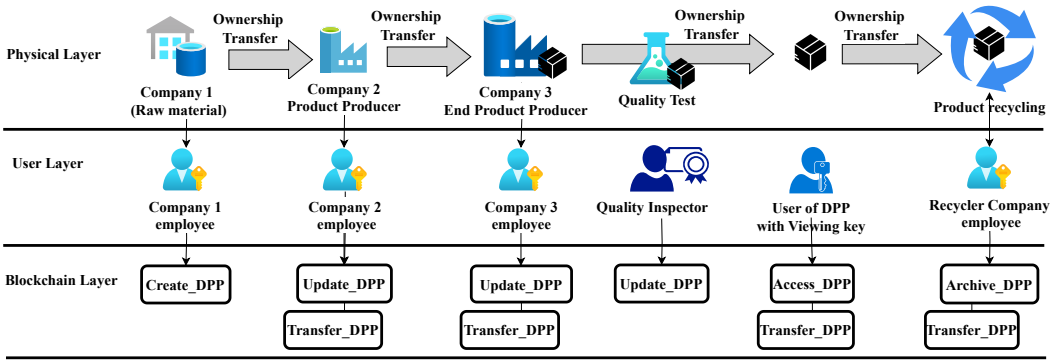


Figure 2.4: Overview of the Blockchain-Enabled DPP Use Case.

2.9 Summary

In this chapter, we illustrated the importance of identity management, access control, anomaly detection, blockchain technologies, and advanced architectures like Zero Trust in ensuring robust IoT security. The dynamic and large-scale nature of IIoT networks demands integrated, automated, and context-aware solutions. Key findings include the distinction between IoT and IIoT, dynamic identity and access management, evolving access control and policy enforcement, and the evolution of ZTA.

Blockchain for IIoT provides decentralization, immutability, and traceability, but scalability, energy consumption, storage overhead, and smart contract execution costs remain barriers to widespread adoption. DPPs enhance product traceability and facilitate circular economy objectives by securely managing lifecycle records. However, interoperability, privacy, and standardization challenges persist.

Network anomaly detection uses various techniques, including knowledge-based, statistical, machine learning, deep learning, and graph-based approaches. However, several critical research gaps remain, such as real-time adaptability, seamless integration, scalability and interoperability, continuous trust evaluation, and universally

accepted metrics for evaluating dynamic Zero Trust implementations and lightweight blockchain solutions.

The chapter sets the stage for subsequent chapters, which aim to address these gaps through novel frameworks and enhancements introduced in the following six chapters. These insights set the stage for further research and development in the field of IIoT security.

Chapter 3

Smart Factories: Security Challenges

Contents

3.1 Introduction

Smart manufacturing uses advanced digital technologies, such as the Industrial Internet of Things (IIoT), to improve industrial automation, efficiency, and responsiveness. IIoT integrates interconnected sensors, controllers, edge devices, and cloud-based analytics to facilitate real-time data exchange, automated decision-making, and improved process visibility across manufacturing operations. Despite these benefits, the rapid adoption and integration of heterogeneous devices and systems in smart manufacturing introduce serval security challenges.

Previous research has explored security aspects of IIoT systems, primarily focusing on individual components, isolated scenarios, or theoretical models. However, comprehensive, system-wide security assessments addressing interactions and dependencies across multiple IIoT tiers remain limited. This chapter addresses this gap by presenting a detailed smart factory use case to systematically identify and analyze vulnerabilities arising from complex device interactions and data flows.

Specifically, this chapter investigates an IIoT communication scenario within a smart factory environment, detailing network architectures, collaborative processes, and interactions between edge devices, Internet of Things (IoT) gateways, and cloud platforms. Using the STRIDE threat modeling approach, we systematically identify vulnerabilities, potential attack vectors, and critical security gaps such as legacy

system vulnerabilities, inadequate real-time monitoring, and data integrity threats. The findings presented in this chapter not only highlight specific areas requiring attention but also inform adaptive mitigation strategies necessary for building resilient and secure smart factory infrastructures.

The contributions of this chapter directly support the thesis's broader objective of developing a robust, security-focused framework for IIoT deployments. By providing a comprehensive security analysis grounded in practical use cases, this chapter enriches the overall understanding of IIoT security risks and sets the foundation for subsequent chapters focusing on threat mitigation, adaptive security measures, and real-time monitoring solutions.

3.2 Use Case Scenario: Smart Factory IIoT Communication

A smart factory integrates IIoT technologies such as industrial sensors, actuators, IoT Gateways, Programmable Logic Controllers (PLCs), cloud-based analytics, and enterprise dashboards to facilitate real-time data collection, automated processes, and informed decision-making. Recent studies emphasize that the adoption of these interconnected technologies significantly improves productivity and operational flexibility; however, it also broadens the attack surface, introducing critical cybersecurity challenges [168, 169, 170]. The smart factory architecture, illustrated in Figure 3.1, highlights the complex interaction among various IIoT components, structured into clearly defined tiers to manage both operational efficiency and security.

The architecture comprises the following tiers:

- **Edge Tier**: Contains sensors, actuators, PLCs, and IoT Gateways. These components collect real-time data, execute localized control decisions, and forward crucial operational data to upper tiers.
- **Platform Tier**: Provides cloud-based services, including data processing, analytics, authentication, and security management. This tier facilitates comprehensive data analysis, ensuring both operational continuity and cybersecurity.
- **Enterprise Tier**: Hosts remote monitoring tools, Supervisory Control and Data Acquisition (SCADA) systems, and analytical dashboards for high-level decision-making and performance monitoring.

3.2.1 Communication and Data Flow in the IIoT System

The smart factory scenario illustrates continuous data exchange among IIoT components, ensuring automation integrity, operational efficiency, and secure remote access. As indicated by recent research [171], communication networks in smart factories, especially when integrating technologies like Software-Defined Networking (SDN), must address significant reliability and cybersecurity issues, including network resilience, latency, and device heterogeneity.

The operational workflow begins with field sensors collecting environmental and production data, which are then processed by PLCs performing immediate logic-based decisions. An IoT Gateway forwards selected data to cloud storage and ana-

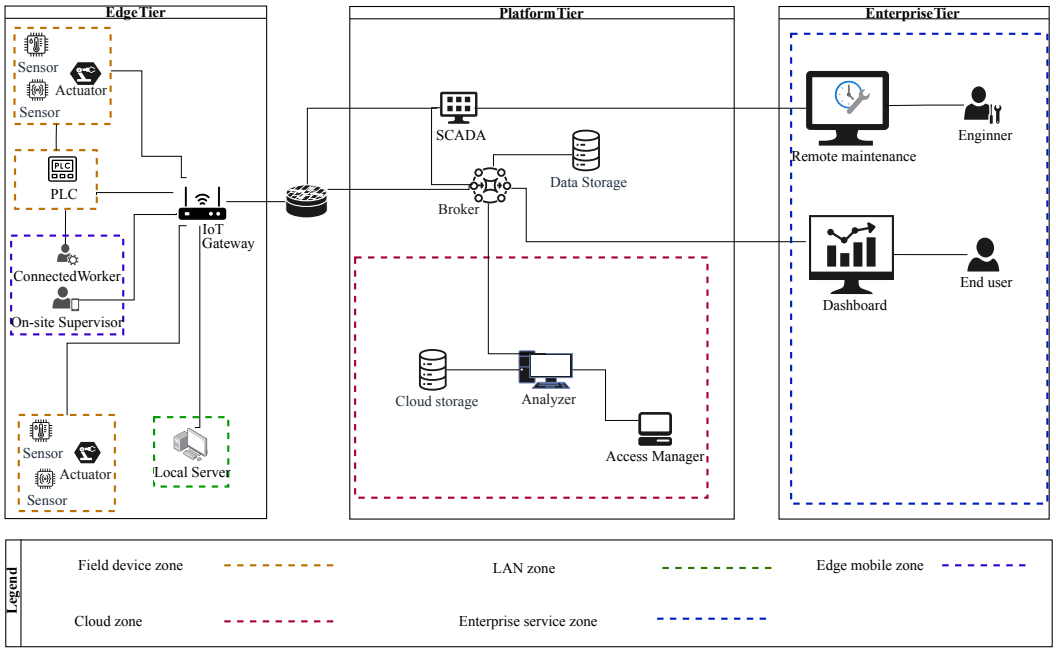


Figure 3.1: Smart Factory IIoT System Architecture.

lytics platforms, satisfying industry demands for real-time responsiveness and robust fault tolerance [168]. Remote maintenance engineers and supervisors use SCADA systems and mobile dashboards to remotely diagnose issues, fine-tune production settings, and ensure continuous operations. Secure cross-zone data exchange among multiple factory sites via IoT gateways further enhances operational consistency and resiliency across distributed production lines.

Recent academic findings highlight specific cybersecurity vulnerabilities introduced by the increased connectivity within these IIoT-driven environments. For instance, traditional security methods such as air-gapping have become ineffective due to highly integrated and interconnected systems, necessitating comprehensive, adaptive security frameworks [169]. Emerging threats, including industrial espionage, data theft, and malicious intrusions, underscore the need for robust, context-aware intrusion detection and proactive defense strategies to secure critical factory operations [172, 173]. Consequently, this use case incorporates multi-tiered, zone-based network segmentation as a core cybersecurity strategy, addressing vulnerabilities identified in recent research.

Entities within each security zone include:

- **Field Device Zone:** PLCs, sensors, actuators, and IoT Gateways managing data collection and local control.
- **LAN Zone:** SCADA systems and local servers providing immediate data processing and storage capabilities.
- **Cloud Zone:** Brokers, access managers, analyzers, and cloud storage ensuring secure remote accessibility, sophisticated data analysis, and robust storage solutions.

- **Enterprise Service Zone**: Dashboards for end-users, remote engineering tools, and centralized storage systems supporting strategic insights and long-term planning.
- **Edge Mobile Zone:** Mobile interfaces and tools utilized by supervisors and connected workers for real-time oversight and control.

3.2.2 Operational Working and Cybersecurity Implications

The continuous operation of a smart factory mandates constant vigilance and proactive management of cybersecurity risks. Real-time sensor data informs operational decisions, while daily and weekly analytics reports underpin strategic assessments and maintenance planning. Recent research identifies significant cybersecurity gaps in current IIoT deployments, particularly the lack of systemic risk understanding, unclear governance structures for cyber incidents, and low cybersecurity awareness among factory personnel [168, 170]. Addressing these concerns requires comprehensive risk-reduction strategies, including enhanced cybersecurity training, clear governance policies, adaptive network defenses, and improved anomaly detection mechanisms based on machine learning [171].

3.3 STRIDE-Based Threat Model for Smart Factory IIoT

We develop the threat model using STRIDE [174] in this section. This model helps systematically identify and mitigate potential security threats across all tiers of the IIoT architecture. We utilize Open Worldwide Application Security Project (OWASP) Threat Dragon, an open-source tool designed for visualizing and analyzing threat models [175]. Threat Dragon assists in mapping data flows, identifying vulnerabilities, and applying STRIDE principles to ensure robust security measures are embedded within the system design.

3.3.1 Identifying Key Components in the Threat Model

To systematically analyze potential threats, the STRIDE-based threat model evaluates security risks across:

- **Actors**, including Remote Engineers, On-Site Supervisors, Connected Workers, and End Users.
- **Processes**, such as Cloud-Based Data Analysis, Remote Access Management, SCADA Processing, and Actuator Control.
- **Data Stores**, including Local Servers, Cloud Storage, and SCADA logs.
- **Data Flows**, such as IoT Gateway Communication, Remote API Access, and Factory Data Sharing.

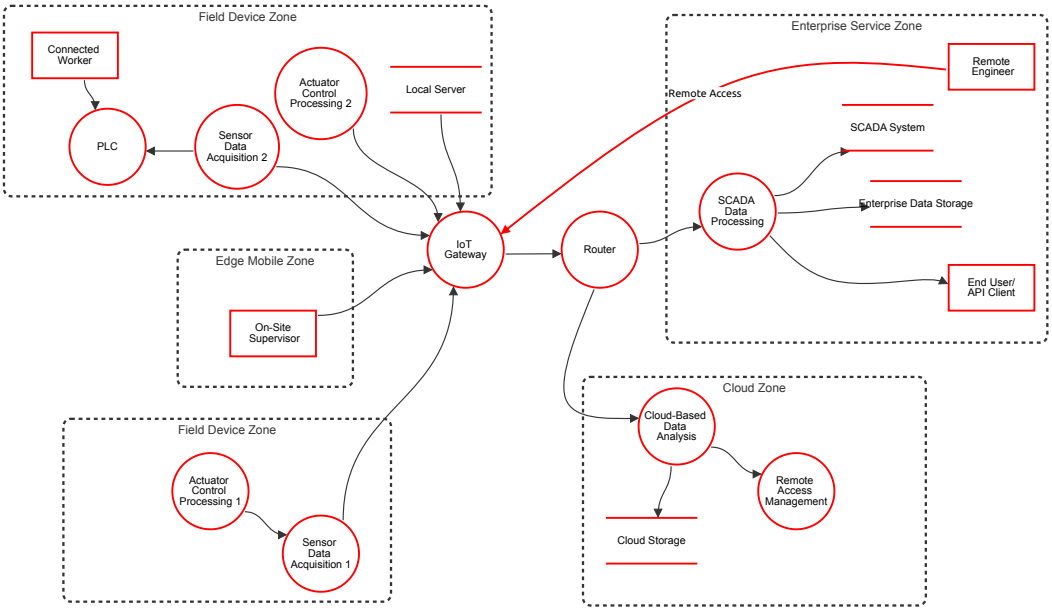


Figure 3.2: Threat Model for Smart Factory IIoT (STRIDE-Based).

3.3.2 STRIDE-Based Threat Analysis

The STRIDE threat model categorizes security risks into Spoofing, Tampering, Repudiation, Information Disclosure, Denial-of-Service (DoS), and Elevation of Privilege (EoP). The following are the primary security risks identified in the smart factory IIoT system:

Spoofing (S) – Unauthorized Identity Use: Attackers may impersonate authorized users, devices, or services to gain control over industrial assets. For example, a hacker may spoof the credentials of a remote engineer to access SCADA systems and modify factory configurations. To mitigate this risk, implementing Multifactor Authentication (MFA), digital certificates Public-Key Infrastructure (PKI), and Zero Trust authentication mechanisms is essential.

Tampering (T) – Data Manipulation: Threat actors may alter sensor readings, PLC commands, or stored logs to disrupt industrial operations. An attacker injecting false temperature data could cause the system to shut down a production line unnecessarily. Countermeasures include data integrity verification (Hash-Based Message Authentication Code (HMAC)), secure data transmission (TLS 1.3), and anomaly detection algorithms.

Repudiation (R) – Denying Unauthorized Actions: Malicious insiders or external attackers may erase logs or modify audit trails to conceal unauthorized activities. A compromised connected worker device might alter SCADA settings and later deny responsibility. To prevent repudiation attacks, blockchain-backed logging and cryptographic event signing should be implemented.

Information Disclosure (I) – Data Leakage: Unauthorized users may gain access to sensitive production data, cloud analytics, or SCADA logs due to weak access controls. Encrypting data in transit and at rest, enforcing Role-Based Access Control (RBAC), and applying network segmentation mitigate these risks.

Denial of Service (D) – Disrupting IIoT Operations: Attackers can launch DoS attacks on IoT gateways, APIs, and cloud storage, causing system failures. Rate limiting, AI-based traffic filtering, and anomaly detection help prevent such disruptions.

Elevation of Privilege (E) – Unauthorized Control: Exploiting misconfigured permissions or unpatched firmware can allow attackers to gain administrative access. Strict RBAC policies, firmware integrity verification, and segmented network controls are necessary defenses.

3.4 Conclusion

This chapter has introduced the use case of Smart Factories within the IIoT, emphasizing the communication mechanisms and data flow integral to their operation. By presenting a representative smart factory scenario, the chapter has underscored the complex interplay between devices, networks, and data exchanges critical for industrial productivity. The detailed STRIDE-based threat model highlights potential vulnerabilities in such interconnected environments, identifying key components and systematically categorizing threats based on Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege.

This use case provides a concrete example of the challenges faced in real-world industrial environments, helping to contextualize the security risks and operational constraints unique to smart factories. Rather than serving as a continuous reference point, the scenario highlights typical vulnerabilities and limitations of existing security models. By grounding abstract concepts in a familiar setting, the chapter motivates the need for dynamic, decentralized, and adaptive security approaches explored throughout the thesis.

Chapter 4

Identity Management, Authentication and Access Policy

Contents

Publications:

- Fatemeh Stodt and Christoph Reich. "Bridge of Trust: Cross Domain Authentication for Industrial Internet of Things (IIoT) Blockchain over Transport Layer Security (TLS)". in: ***Electronics*** 12.11 (2023), p. 2401
- Fatemeh Stodt and Christoph Reich. "A Review on Digital Wallets and Federated Service for Future of Cloud Services Identity Management". In: ***SERVICE COMPUTATION 2023, The Fifteenth International Conference on Advanced Service Computing, June 26-30, Nice, France***. IARIA. 2023 **(Best paper award)**
- Fatemeh Stodt and Christoph Reich. "Digital Wallets and Identity Management: Pioneering Advances for Cloud Service Evolution". In: ***International Journal on Advances in Software*** 17.1 (2024), pp. 13–22

4.1 Introduction

Identity management is a foundational pillar of cybersecurity in IIoT environments. As smart factories evolve into highly connected and decentralized ecosystems, ensuring that only authorized devices and users can access critical industrial assets becomes increasingly complex. Traditional Identity and Access Management (IAM) models often rely on centralized infrastructures, which are ill-suited to the scalability, interoperability, and dynamism of distributed IIoT networks.

In such environments, effective identity management must address not only authentication and access control but also enable trust across independently managed domains. Furthermore, these mechanisms must be lightweight and adaptable to meet the resource constraints of IIoT devices while maintaining high security and responsiveness.

To tackle these challenges, this chapter introduces two complementary contributions that collectively enhance identity management and authentication in dynamic IIoT settings:

Digital Wallet-Based Identity Management (Section 4.2): This contribution introduces a decentralized identity framework tailored for IIoT networks, leveraging digital wallets to manage and verify device identities efficiently. Unlike earlier approaches based on Hyperledger Indy or Ethereum, which incur high computational and operational costs, our design allows most verifications to occur off-chain. It categorizes devices by security needs and supports scalable identity management, ensuring interoperability and minimal overhead across industrial subsystems.

While digital wallet-based identity management offers a secure and decentralized approach to credential handling, it faces challenges when devices must communicate across different organizational boundaries. Many IIoT deployments span multiple industrial domains, each governed by its own trust rules, requiring seamless interoperability. To complement the strengths of digital wallets and address these cross-organizational requirements, this chapter also introduces a cross-domain authentication framework. Together, these two contributions form an integrated, end-to-end solution that tackles both local identity management and broad inter-domain trust in distributed IIoT ecosystems.

Cross-Domain Authentication for IIoT (Section 4.3): Addressing the challenge of trust across organizational boundaries, this contribution presents a blockchain-enhanced authentication protocol integrated with Transport Layer Security (TLS). Prior approaches, such as the accumulator-based schemes, provide strong revocation guarantees but suffer from high complexity and latency. Our solution avoids such pitfalls by enabling fast, secure, and scalable cross-domain authentication suitable for real-time IIoT interactions.

The originality of this chapter lies in integration of decentralized identity management and blockchain-enabled authentication into a unified, scalable framework. By combining off-chain identity verification, role-aware access policies, and interoperable authentication, the proposed solutions enhance resilience, reduce latency, and support automation across diverse industrial environments – key features for securing next-generation smart factory networks.

Addresses Research Questions:

- **How can scalable and secure identity management be achieved in distributed IIoT networks?**

4.2 Advanced Digital Wallet Identity Management for IIoT

IIoT environments require identity management solutions that securely and efficiently handle device enrollment, authentication, and credential lifecycle management. Traditional methods often struggle with scalability and centralized points of failure.

This research specifically addresses these challenges by introducing a digital wallet-based identity management architecture tailored to IIoT. Digital wallets offer decentralized trust mechanisms, notably blockchain and federated identity protocols, ensuring robust authentication, enhanced encryption, and efficient credential management across diverse IIoT devices.

This section explicitly outlines the specific identity management challenges encountered in IIoT, proposes a detailed device categorization framework, and presents a novel architecture leveraging digital wallets to securely manage device identities throughout their lifecycle.

Existing studies on digital wallet-based identity management for IIoT have explored various approaches but suffer from significant limitations in scalability, security, and adaptability. Regueiro *et al.* [176] and Dixit *et al.* [177] rely on Hyperledger Indy and Ethereum-based decentralized identity models, which introduce high computational overhead and scalability issues due to their dependence on multiple blockchain networks and off-chain storage layers.

Sahmim *et al.* [178] propose an edge-based identity wallet, improving local authentication but lacking cross-domain interoperability and remaining vulnerable to edge node compromises. Popa *et al.* [179] introduce ChainDiscipline, a multi-domain decentralized identity model with trust scoring, but its complex blockchain consensus mechanisms make it impractical for real-time IIoT operations.

In contrast, our hierarchical digital wallet-based identity management framework addresses these challenges by reducing blockchain dependency and handling most identity verifications off-chain. It categorizes IIoT devices based on their security needs, enabling more targeted and efficient identity management.

Unlike previous works that rely on specific blockchain implementations (e.g., Hyperledger or Ethereum), our approach is ledger-agnostic, ensuring greater interoperability and adaptability. Additionally, by integrating hardware-backed credentials (Trusted Platform Module (TPM)/Hardware Security Module (HSM)) for high-security devices, our framework enhances resilience against attacks, even if edge nodes are compromised

This adaptive and scalable architecture provides real-time authentication, decentralized trust, and seamless integration with industrial ecosystems, making it a more practical and future-proof solution for IIoT identity management.

4.2.1 Expected Properties

Effective identity management for IIoT must address several critical requirements [180], as listed:

- **R1 – Secure Storage of Identity-Related Data:** Ensuring that IIoT credentials remain protected from unauthorized access, breaches, or tampering. Secure storage mechanisms rely on strong cryptography, rigorous access controls, and continuous monitoring [181].
- **R2 – Effective Management of Identity-Related Data:** Beyond secure storage, identity data must be properly managed throughout its lifecycle. This includes credential issuance, authentication, revocation, updates, and policy enforcement to maintain security and compliance in dynamic IIoT environments [182].
- **R3 – Secure Sharing of Identity-Related Data:** Identity-related data must be protected when transmitted across factories, partners, or cloud services. Secure communication protocols and encryption techniques prevent unauthorized interception [183].
- **R4 – Secure Storage of Cryptographic Material:** Protecting cryptographic keys and certificates is essential for device authentication and secure communication. Tamper-resistant hardware modules (e.g., TPM, HSM) can help mitigate key exposure risks [184].
- **R5 – Combining Identity Data Before Sharing:** Administrators must be able to merge or split identity attributes based on operational needs. This ensures privacy, access control, and compliance with data-sharing policies [185].

4.2.2 Proposed Architecture

The Identity Management Service (IMS) is the central entity in the framework, responsible for issuing, updating, and revoking digital credentials. It ensures device credentials remain valid throughout their lifecycle. A blockchain-based ledger within the Federated Layer records all identity-related events, enhancing trust and system resilience. The Application and Service Layer enforces fine-grained access control across various industrial applications, ensuring only devices with validated credentials can access sensitive resources, maintaining strict security standards.

Effective access management is achieved by categorizing IIoT devices based on their security and privacy requirements. As illustrated in Figure 4.1, devices can be grouped into three categories:

1. **Low-Security Devices:** Environmental sensors handle non-critical data with lightweight credentials and basic encryption, ensuring minimal identity attributes are maintained in a digital wallet due to the low risk associated with transit data.
2. **Moderate-Security Devices:** Moderately sensitive devices like production line controllers and supervisory systems utilize enhanced security measures like

multifactor authentication and certificate-based methods, along with encryption, for better access control. Their digital wallets contain richer credential sets, allowing for granular control.

3. **High-Security Devices:** Mission-critical devices like safety interlocks require high-level protection through hardware-backed credentials, continuous monitoring, and advanced authentication protocols. Digital wallets are equipped with immutable credentials and integrated with real-time risk assessment tools for enhanced security.

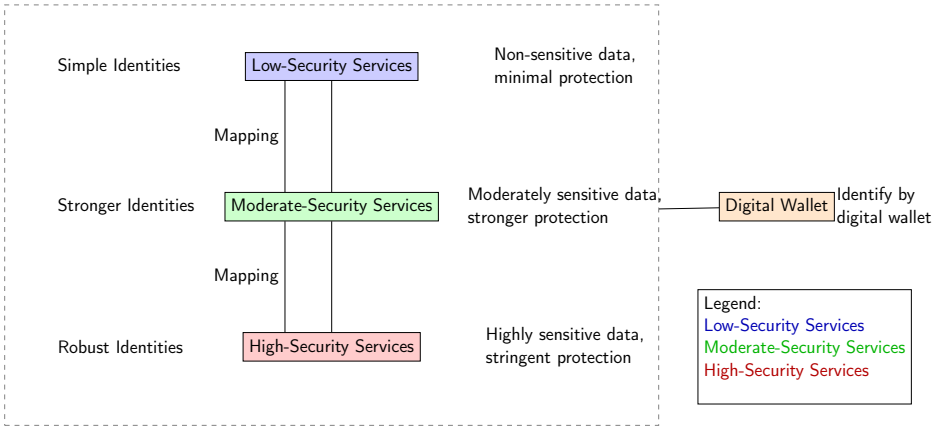


Figure 4.1: Categorization of IIoT Devices and Identity Types.

This tiered approach enables the system to apply security mechanisms proportionate to device criticality while maintaining operational efficiency. Overall, the lifecycle of identity management in this framework is divided into three primary phases: enrollment, authentication, and credential management (which includes updates and revocations). Each phase is supported by dedicated algorithms that interact seamlessly, ensuring that the system remains both adaptive to evolving threats and efficient in its operation.

Phase 1: Enrollment and Credential Issuance Algorithm 1 outlines the secure enrollment process for IIoT devices, ensuring strong authentication and tamper-resistant credential storage.

First, the device undergoes hardware-backed attestation using a TPM verified device identity key (Line 1), which is validated by the Edge Node to confirm the device's integrity (Line 2). If the attestation fails, enrollment is immediately rejected (Lines 3-4) to prevent unauthorized device access. Upon successful attestation, the device proceeds to registration with the IMS (Line 6).

The IMS then assigns a Decentralized Identifier (DID) and issues a Verifiable Credential (VC), cryptographically binding the device identity (Lines 7-8). These credentials are securely stored in the device's digital wallet (Line 9), ensuring persistent and verifiable authentication. Finally, an immutable record of the enrollment event is written to the blockchain (Line 10), enhancing auditability and trust across federated IIoT environments.

Algorithm 1: Enrollment and Credential Issuance

Data: New IIoT Device
Result: Device Registered with Credentials in Digital Wallet
1. Device Attestation:
    Device sends attestation evidence (e.g., TPM certificate) to the local Edge Node;
    Edge Node verifies the device's secure hardware integrity;
    if *Attestation fails* then
        Reject enrollment and exit;
    else
        Proceed to registration;
    end
2. Registration with IMS:
    Device registers with the Identity Management Service (IMS);
    IMS assigns a unique Decentralized Identifier (DID) and issues Verifiable Credentials (VCs) based on device attributes;
    Device securely stores the issued credentials in its Digital Wallet using hardware-backed security;
    Record the enrollment event on the Blockchain Ledger for auditability;
return Device successfully enrolled;

This layered approach mitigates spoofing, unauthorized access, and credential tampering, providing a scalable and resilient identity management system for IIoT deployments.

Phase 2: Authentication and Access Control Algorithm 2 defines the authentication and access control process, ensuring that only verified IIoT devices can access network services.

The device initiates an access request using mutual TLS (mTLS) and signs it with credentials from its digital wallet (Lines 1-2). The Edge Node then validates the device's attestation evidence and stored credentials (Line 4), preventing unauthorized devices from gaining access. To ensure registration legitimacy, the Edge Node queries the blockchain ledger (Line 5), confirming the device's prior enrollment and status.

Next, the IMS retrieves the device credentials from the digital wallet (Line 7) and evaluates them against security policies and contextual risk factors (Line 8). If the authentication and policy evaluation pass, the device is granted access (Line 9); otherwise, access is denied (Line 10).

By integrating off-chain verification with blockchain-backed auditability, this adaptive access control model ensures low-latency authentication, resistance to credential spoofing, and dynamic risk-aware authorization in IIoT environments.

Phase 3: Credential Update and Revocation Algorithm 3 outlines the credential update and revocation process, ensuring that the system remains resilient to compromised or misbehaving devices.

The Edge Nodes and IMS continuously monitor device behavior and environmental context (Line 1). If an anomaly or policy violation is detected, the system triggers a credential update or revocation (Line 2), preventing unauthorized access.

Algorithm 2: Authentication and Access Control

Data: Access Request from an IIoT Device
Result: Access Granted or Denied
1. Access Request Initiation:
  Device initiates an access request using mutual TLS (mTLS);
  Device signs the request with credentials from its Digital Wallet;
2. Edge Node Validation:
  Edge Node validates the attestation evidence and Digital Wallet credentials;
  Edge Node queries the Blockchain Ledger to verify the device's registration status;
3. Context-Aware Authorization:
  IMS retrieves the device's credentials from the Digital Wallet;
  IMS evaluates the credentials against current security policies and risk context using smart contracts;
  if ***Evaluation passes*** **then**
    Grant access to the requested service;
  else
    Deny access;
  end
return Access status based on evaluation;

Algorithm 3: Credential Update and Revocation

Data: Monitoring Alerts from Edge Nodes/IMS
Result: Updated Credential State in Digital Wallet and Blockchain Ledger
1. Continuous Monitoring:
  Edge Nodes and IMS monitor device behavior and environmental context;
  If an anomaly or policy violation is detected, trigger the update/revocation process;
2. Credential Update/Revocation:
  IMS initiates an update or revocation of the device's credentials stored in the Digital Wallet;
  Record the update or revocation event on the Blockchain Ledger;
  Propagate the change across federated networks to ensure global consistency;
return Updated device credential status;

Upon detection, the IMS updates or revokes the device's credentials stored in the digital wallet (Line 4) and records the event on the blockchain ledger (Line 5) to ensure auditability and tamper-proof tracking. The update is then propagated across federated networks (Line 6) to maintain consistency in distributed identity verification.

By dynamically adapting credentials based on real-time risk assessment, this mechanism prevents unauthorized access, mitigates insider threats, and maintains trust across IIoT networks without disrupting legitimate operations.

4.2.3 Security Proof

In this section, we present a formal proof of the security properties of our proposed digital wallet-based identity management framework. Our proof relies on two key results: (i) a reduction showing that any successful forgery of device credentials

would contradict the assumed security of the underlying digital signature scheme, and (ii) an inductive argument demonstrating that the system's security invariant is maintained over time.

Definition 1 (Digital Signature Scheme)**.** *Let* $\Sigma = (\mathsf{KG}, \mathsf{Sign}, \mathsf{Verify})$ *be a digital signature scheme, where:*

$$\mathsf{KG}(1^\lambda) \to (sk, pk),$$

$$\mathsf{Sign}(sk, m) \to \sigma,$$

$$\mathsf{Verify}(pk, m, \sigma) \in \{0, 1\}.$$

We assume that Σ *is existentially unforgeable under chosen-message attacks (EUF-CMA); that is, for any probabilistic polynomial-time adversary* $\mathcal{A}$*, the probability of producing a pair* (m^*, σ^*) *such that*

$$\mathsf{Verify}(pk, m^*, \sigma^*) = 1,$$

with m^* *not queried to the signing oracle, is negligible.*

Definition 2 (Digital Credential)**.** *A digital credential for a device* D *is defined as the tuple:*

$$C = (D, m, \sigma),$$

where m *contains identity and attribute information for* D*, and* $\sigma = \mathsf{Sign}(sk, m)$ *is the signature binding the information to* D*.*

Definition 3 (Enrollment)**.** *A device* D *is considered* **enrolled** *if it passes hardware-backed attestation and is issued a credential* $C = (D, m, \sigma)$ *by the Identity Management Service (IMS). We denote the enrollment process by:*

$$Enroll(D) \to C,$$

which also guarantees that the credential is stored securely in D*'s digital wallet.*

Definition 4 (System State and Invariant)**.** *Let* $\mathcal{S}_t$ *denote the state of the system at time* t*, including all devices and their associated credentials. We define the security invariant* $I(t)$ *as:*

$$I(t): \quad \forall\, D \in \mathcal{S}_t,\ D \text{ is enrolled with a valid credential } C = (D, m, \sigma), \text{ and } \mathsf{Verify}(pk, m, \sigma) = 1.$$

Theorem 1: Credential Unforgeability via Reduction

Theorem 1. *Under the assumption that* Σ *is EUF-CMA secure, no probabilistic polynomial-time adversary* $\mathcal{A}$ *can produce a valid forged credential*

$$C^* = (D^*, m^*, \sigma^*)$$

such that $\mathsf{Verify}(pk, m^*, \sigma^*) = 1$*, unless* m^* *was generated by a legitimate execution of the enrollment procedure.*

Proof. Assume, for contradiction, that there exists an adversary $\mathcal{A}$ that can output a forged credential $C^* = (D^*, m^*, \sigma^*)$ with non-negligible probability. We construct an algorithm $\mathcal{B}$ that uses $\mathcal{A}$ as a subroutine to forge a signature under Σ as follows:

1. $\mathcal{B}$ receives the public key pk from the digital signature challenger and is given oracle access to a signing function $\mathsf{Sign}(sk, \cdot)$.

2. $\mathcal{B}$ simulates the enrollment process for $\mathcal{A}$ by answering any signing queries using the signing oracle.

3. Eventually, $\mathcal{A}$ outputs a forged credential $C^* = (D^*, m^*, \sigma^*)$ such that $\mathsf{Verify}(pk, m^*, \sigma^*) = 1$ and m^* was never queried.

By the EUF-CMA security of Σ, the probability that $\mathcal{B}$ successfully forges a signature is negligible. This contradicts our assumption that $\mathcal{A}$ can forge C^* with non-negligible probability. Hence, no such adversary exists, and the unforgeability of credentials is assured. □

Theorem 2: Maintenance of the Security Invariant via Induction

Theorem 2. *The system's security invariant $I(t)$ is maintained for all time steps $t \geq 0$.*

Proof. We prove this by induction on the time steps.

Base Case: At time $t = 0$, the system initializes with the enrollment process:

$$Enroll(D) \rightarrow C = (D, m, \sigma),$$

for every device D that is enrolled. By construction, $\mathsf{Verify}(pk, m, \sigma) = 1$, hence $I(0)$ holds.

Inductive Step: Assume that at time t, the invariant $I(t)$ holds, i.e., every device $D \in \mathcal{S}_t$ is enrolled with a valid credential. At time $t+1$, one of the following operations occurs:

1. **Authentication:** When a device D attempts to authenticate (see Algorithm 2), its credential $C = (D, m, \sigma)$ is re-verified by both the Edge Node and the blockchain ledger. If the credential fails verification, access is denied and a credential update/revocation is triggered, ensuring that only devices with valid credentials remain active.

2. **Credential Update/Revocation:** When a credential update or revocation occurs (see Algorithm 3), the IMS enforces that any change is recorded and that compromised or invalid credentials are removed from the active set $\mathcal{S}_{t+1}$.

In either case, the system ensures that every device in $\mathcal{S}_{t+1}$ satisfies:

$$\mathsf{Verify}(pk, m, \sigma) = 1.$$

Thus, $I(t+1)$ holds.

By mathematical induction, the security invariant $I(t)$ holds for all $t \geq 0$. □

Theorem 1 shows that any forgery of a digital credential directly contradicts the Existential Unforgeability under Chosen Message Attack (EUF-CMA) security of the underlying digital signature scheme. Theorem 2 establishes that the system's security invariant, ensuring all active devices possess valid and verifiable credentials,

is maintained over time. Together, these results mathematically validate the correctness and robustness of the proposed digital wallet-based identity management framework for IIoT.

In conclusion, digital wallets significantly enhance security and scalability by decentralizing identity and credential management. Yet, in many IIoT deployments, devices must engage with services managed by other entities or consortia. Relying solely on digital wallets does not fully address the complexities of establishing trust and secure interactions across multiple administrative domains. To fill this gap, the next section presents an adaptable cross-domain authentication method that extends these digital wallet principles, ensuring that verified identities can operate safely across independently managed networks.

4.3 Cross-Domain Authentication

Building on the secure foundation provided by digital wallet-based identity management, the focus now shifts to cross-domain authentication essential for scenarios where devices and services belong to separate administrative realms. While digital wallets capably handle lifecycle management and credential issuance, cross-domain interoperability demands an additional trust infrastructure. This section thus introduces a blockchain-backed authentication scheme that bridges the gap between isolated trust domains, offering a low-latency, highly secure mechanism for verifying identities and permissions in complex IIoT settings. In Figure 4.2, we see the problem model where multiple enterprises and organizations share resources in a decentralized network environment. In this scenario, multiple enterprises and organizations share resources, requiring multiple domains for services. A single trust domain cannot provide numerous services, requiring users to visit multiple domains. For example, a user from domain A needs to access a service in domain B, which requires obtaining their root Certificate Authority (CA) certificate. This method has drawbacks like complex authentication, frequent signature verification, and certificate management complexity. Alternatively, domains can be certified by a third-party body, but this can create single points of failure and privacy breaches.

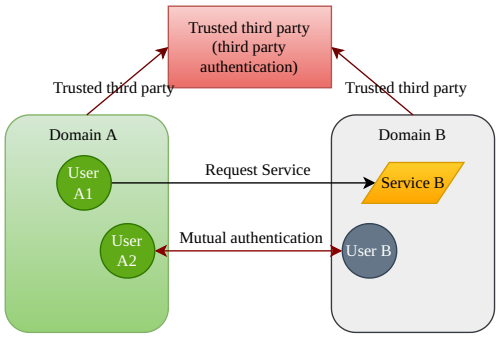


Figure 4.2: Cross Domain Problem Model.

Cross-domain authentication is crucial in IIoT environments, enabling secure and efficient interaction among heterogeneous devices and services across multiple administrative or operational domains. Existing approaches primarily leverage

blockchain technologies, cryptographic accumulators, or hardware-based security mechanisms to enhance authentication efficiency, security, and scalability. However, these solutions have limitations impacting their practicality for real-time industrial deployments.

Gao *et al.* [186] propose a blockchain-based identity management framework using pseudonymous identities, maintained in decentralized ledgers for each domain. While enhancing privacy, their solution incurs high computational overhead due to frequent blockchain transactions, leading to processing delays and limited scalability. Similarly, Cui *et al.* [187] introduce an anonymous authentication approach using dynamic cryptographic accumulators for efficient revocation and identity updating. Despite improved security and anonymity, their solution still experiences delays from computationally intensive cryptographic operations, unsuitable for real-time industrial scenarios.

Wang *et al.* [188] propose a blockchain-based lightweight message authentication framework leveraging Elliptic-Curve Cryptography (ECC) and edge-assisted authentication to minimize computational overhead. Although performance improves, dependency on edge nodes introduces potential single points of failure, impacting overall security and reliability. Khalid *et al.* [189] utilize blockchain combined with Physically Unclonable Function (PUF)-based keys for unclonable device identities. Despite reduced computational requirements, their method mandates specialized hardware compatibility, restricting scalability and posing risks related to key degradation or recovery.

The secure exchange of data between field devices is critical for ensuring the integrity and confidentiality of industrial communication networks. Field devices, typically composed of a field bus application and a communication module, operate in automation networks that often lack built-in security mechanisms. Existing solutions usually focus on securing communication within local field bus environments, yet cross-network communication, especially via the internet, requires additional mechanisms to establish and maintain trust and data integrity. Traditional methods often rely on TLS; however, directly applying TLS to field bus networks introduces challenges, such as non-IP-based communication constraints and address translation complexities.

In contrast to these existing approaches, our proposed solution addresses these identified shortcomings by introducing a practical and scalable TLS-based cross-domain authentication mechanism. Unlike blockchain-based or cryptographic accumulator approaches, our method significantly reduces authentication latency, achieving real-time responsiveness essential for IIoT. It avoids reliance on edge nodes or specialized hardware, eliminating associated security vulnerabilities and scalability limitations. This makes our solution broadly applicable and readily deployable in diverse real-world industrial settings.

The following subsections detail our assumptions and technical requirements, cross-field bus communication and authentication models, and the formal proof validating our approach.

4.3.1 Assumptions and Requirements

Cross-domain authentication in industrial networks, especially for traditional systems such as SCADA, is critical due to inherent vulnerabilities and exposure to diverse security threats [190]. Authentication plays a central role in network security by verifying device and user identities, thus limiting communication strictly to authenticated entities and processes. This prevents unauthorized access and ensures secure interactions between field devices utilizing industrial protocols [191].

While previous sections introduced general assumptions and requirements for identity and access management, the context of cross-domain communication introduces specific assumptions and technical requirements. Below, we explicitly clarify the assumptions considered true in our approach and outline distinct technical requirements necessary for implementing secure cross-domain authentication.

We assume the following conditions hold true in the industrial environment under consideration:

- **Existence of Internal Field Bus Communication:** Native communication between field devices and integrated controllers is already established and functional within each domain.
- **Presence of a Local Public Key Infrastructure (PKI):** Each domain has a functioning internal PKI limited specifically to the field bus area, managing device keys and certificates securely throughout the lifecycle of each device.
- **Existence of Trusty Full Nodes:** Each domain has trustworthy blockchain nodes (full nodes) responsible for blockchain authentication management, device management, and maintaining a trusted table of neighbor nodes.

Also, to realize effective cross-domain authentication, the following requirements must be fulfilled:

1. **End-to-end Secure Communication:** Communication between two distinct domain endpoints must be secured through cryptographic keys or certificates for mutual authentication.
2. **PKI Extension or Integration:** The internal domain-specific PKI must be capable of secure extension or integration into cross-domain authentication scenarios. It must enforce strict monitoring to prevent unauthorized direct connections from external domains.
3. **Blockchain Management Nodes:** Each participating domain requires blockchain management nodes (trusty full nodes) to manage authentication and facilitate blockchain operations within and between domains.
4. **Trusty Neighbor Nodes Table:** Each blockchain node must maintain an updated and trusted list of neighboring nodes to ensure robust cross-domain blockchain synchronization and authentication.
5. **Use of TLS (Transport Layer Security):** We specifically require using the TLS protocol for securing cross-domain communication due to its proven properties: it has minimal dependency on adjacent communication layers,

ensuring broad applicability; it demonstrates maturity through widespread deployment and intensive security testing; it offers high adaptability to diverse operational requirements through configurable parameters; and it does not rely strictly on IP-based networks, requiring only targeted packet delivery (TLS records) capabilities.

These assumptions and requirements specifically address cross-domain aspects and complement previously defined general security assumptions. They ensure robust security mechanisms tailored explicitly for industrial cross-domain scenarios.

The cross-domain authentication mechanism developed reuses the Trust Management System (TMS) originally introduced in the "Schloss" architecture [192], a prior publication by the author. The TMS leverages a game-theoretic reputation model to dynamically assess node trustworthiness, facilitating decentralized trust decisions based on feedback and behavioral history. Within the current framework, this component is incorporated without modification to support cross-domain authorization and ensure secure synchronization across industrial networks.

4.3.2 Cross-Field Bus Communication Model

Our cross-field bus communication model is specifically designed to address the identified security and scalability limitations for cross-domain communication. The model assumes a structured communication pattern in which field devices communicate through blockchain-managed authentication gateways. Each domain maintains internal field bus communication secured by local cryptographic mechanisms, while cross-domain interactions are facilitated securely by leveraging a blockchain-managed TLS-secured communication channel.

To enable secure and transparent cross-domain interactions, our model integrates dedicated authentication gateways at each domain boundary. These gateways manage TLS sessions, handle secure address translations between domains, and interface seamlessly with the blockchain-based identity verification system, ensuring robust authentication and secure session management without disrupting native field bus communication.

Figure 4.3 illustrates our cross-field bus communication model architecture, clearly highlighting the role of blockchain nodes, domain-specific gateways, and secure inter-domain communication channels.

4.3.3 Cross-Field Bus Authentication Model

In our approach, we aim to achieve secure, scalable, and reliable authentication across field bus domains. Specifically, our authentication model focuses on:

- **Robust Identity Verification:** Ensuring the identities of communicating entities are authentic and verifiable.
- **Dynamic Trust Establishment:** Allowing devices from different subnetworks or domains to authenticate securely, even in environments with limited direct certificate validation capabilities.

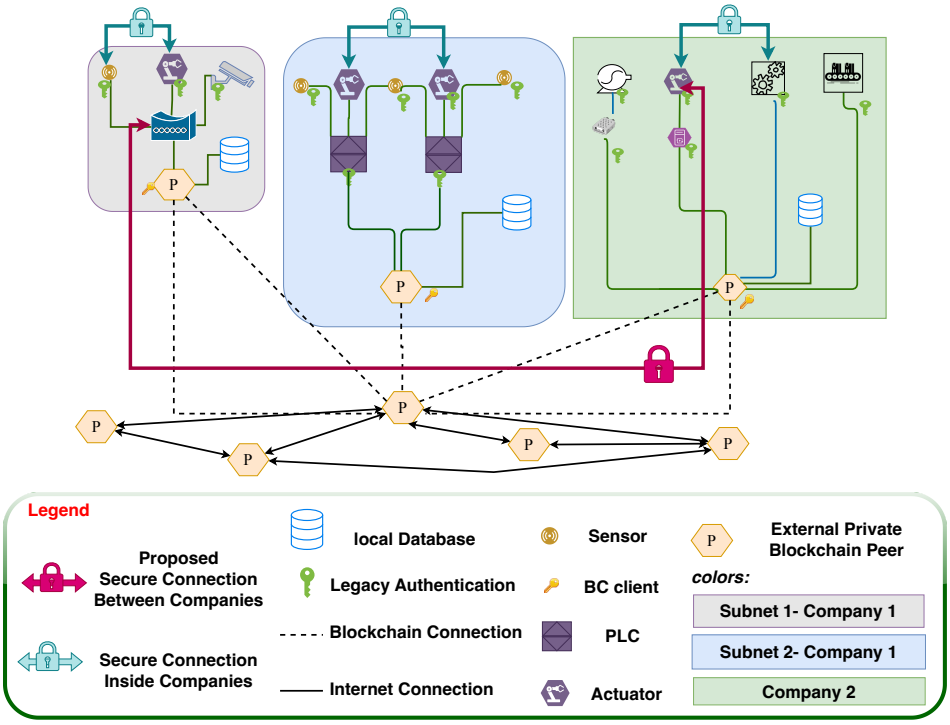


Figure 4.3: Cross Authentication Between Different Subnets.

- **Reduced Computational Overhead:** Minimizing computational and communication costs inherent to traditional certificate validation processes, especially important for resource-constrained industrial devices.
- **Resilience to Network Segmentation:** Enabling secure cross-domain communication despite network segmentation or address translation limitations.

Our proposed model utilizes X.509 certificates widely recognized for verifying identities and establishing secure communication in TLS-based exchanges [193]. While effective, traditional TLS certificate validation introduces significant overhead, especially during the handshake phase involving computationally intensive operations like certificate validation and key exchanges [194]. These overheads pose challenges for real-time communication within resource-constrained IIoT environments.

To address these limitations, our authentication model integrates a blockchain-based trust management mechanism alongside traditional X.509 certificates. This hybrid solution significantly reduces dependence on centralized CAs and lengthy certificate chains by dynamically managing trust through decentralized blockchain nodes.

4.3.3.1 Intuitive Idea and Originality

Instead of relying exclusively on hierarchical certificate validation (traditional PKI), our model uses blockchain-managed trust lists to dynamically and efficiently establish cross-domain trust. Blockchain nodes maintain decentralized trust value lists representing device trustworthiness, enabling immediate authentication decisions

Table 4.1: Description of Symbols.

Description	Parameter
Device in domain A	D_A
Peer in domain A	P_A
Device in domain B	D_B
Peer in domain B	P_B
Trust value device i	T_i
Blockchain	BC
Certificate	Cert
Signature	Sig

even in environments lacking direct TLS connectivity. This unique integration of blockchain-based decentralized trust with traditional X.509-based authentication introduces practical benefits including reduced latency, increased resilience to node or network failures, and higher scalability in complex industrial deployments.

4.3.3.2 Detailed Description and Justifications

The authentication workflow combines X.509 certificates with blockchain-managed trust validation as follows:

- Each device in a domain (e.g., D_A in domain A and D_B in domain B) possesses a unique X.509 certificate (Cert) initially issued by the domain's internal CA.
- Blockchain nodes (BC) maintain dynamic trust values (T_i) associated with each registered device. These trust values reflect historical behavior, device legitimacy, and policy compliance over time.
- When devices from different domains initiate communication, they first attempt a direct TLS handshake using their respective X.509 certificates. If network constraints (segmentation or address translation issues) prevent direct certificate validation, blockchain nodes provide a decentralized verification of device trustworthiness based on stored trust values (T_i).
- The blockchain-based trust values allow devices to authenticate securely by confirming each other's legitimacy without direct CA involvement or complete certificate chain verification, significantly reducing handshake complexity.

4.3.3.3 Trust Management Policies

In this context, "trust management policies" refer explicitly to predefined security rules encoded in blockchain smart contracts. These policies specify conditions under which trust values (T_i) are updated, managed, and used for authentication decisions, ensuring compliance with organizational security standards and requirements.

Figure 4.4 illustrates the detailed authentication workflow. Table 4.1 provides a concise summary of the symbols and parameters involved.

The authentication process aims to establish a secure and trusted connection between two devices, D_A and D_B, across different domains while ensuring compliance with trust management policies. This process consists of multiple steps to

validate the legitimacy of the devices, verify trustworthiness, and enable encrypted communication.

1. **Validation of Registration Time and Trustworthiness:** D_A initiates the authentication process by calling the function `Valid(`D_B`)`, which checks whether D_B's registration time in domain B is still valid. This step is necessary to prevent unauthorized or expired devices from participating in the network. Additionally, P_A, acting on behalf of D_A, queries the consortium blockchain to retrieve trust values associated with D_B. The returned result contains P_B's identity and its trust value T_i. If T_i does not meet the required threshold or is invalid, P_A refuses the request from D_A to connect with D_B, terminating the session (steps 1–4 in Figure 4.4).

2. **Trust Evaluation of the Requesting Device:** Once D_B has been validated, the next step involves verifying P_A's trustworthiness. P_A requests permission from P_B to establish a connection with D_B. Before granting access, P_B evaluates whether P_A itself is trustworthy by checking the blockchain-stored trust values. This verification step is crucial because it ensures that only trusted entities can participate in authentication, thereby preventing unauthorized access. Based on the obtained trust value, P_B decides whether to continue the authentication process or terminate the session if P_A does not meet the trust requirements (steps 5–8 in Figure 4.4).

3. **Final Approval and Certificate Signing:** If the authentication proceeds, P_B then requests P_A to validate D_A by signing its certificate ($\text{cert}(D_A)$). This step establishes cryptographic proof of identity, ensuring that D_A is not an impersonator or a rogue device. Once P_A signs and returns the certificate, D_A is officially approved to connect with D_B. This stage prevents unauthorized devices from gaining access to the system and ensures that every participant in the communication is authenticated through verifiable credentials (steps 9–21 in Figure 4.4).

4. **Mutual TLS Handshake for Secure Communication:** Following successful authentication, a secure session must be established between D_A and D_B to enable confidential communication. To achieve this, D_B initiates the *TLS handshake*, transmitting a temporary public key to D_A. This handshake is essential as it negotiates encryption parameters and ensures data integrity. D_A responds by sending its own *TLS handshake message*, completing the key exchange process and confirming mutual agreement on encryption standards (steps 22–23 in Figure 4.4).

5. **Establishing a Secure Session:** Finally, once the TLS handshake is complete, a secure and encrypted communication channel is established between D_A and D_B. From this point onward, all exchanged information remains protected from eavesdropping, ensuring data confidentiality and integrity. The successful authentication and secure session setup allow D_A and D_B to communicate across different domains without compromising security (step 24 in Figure 4.4).

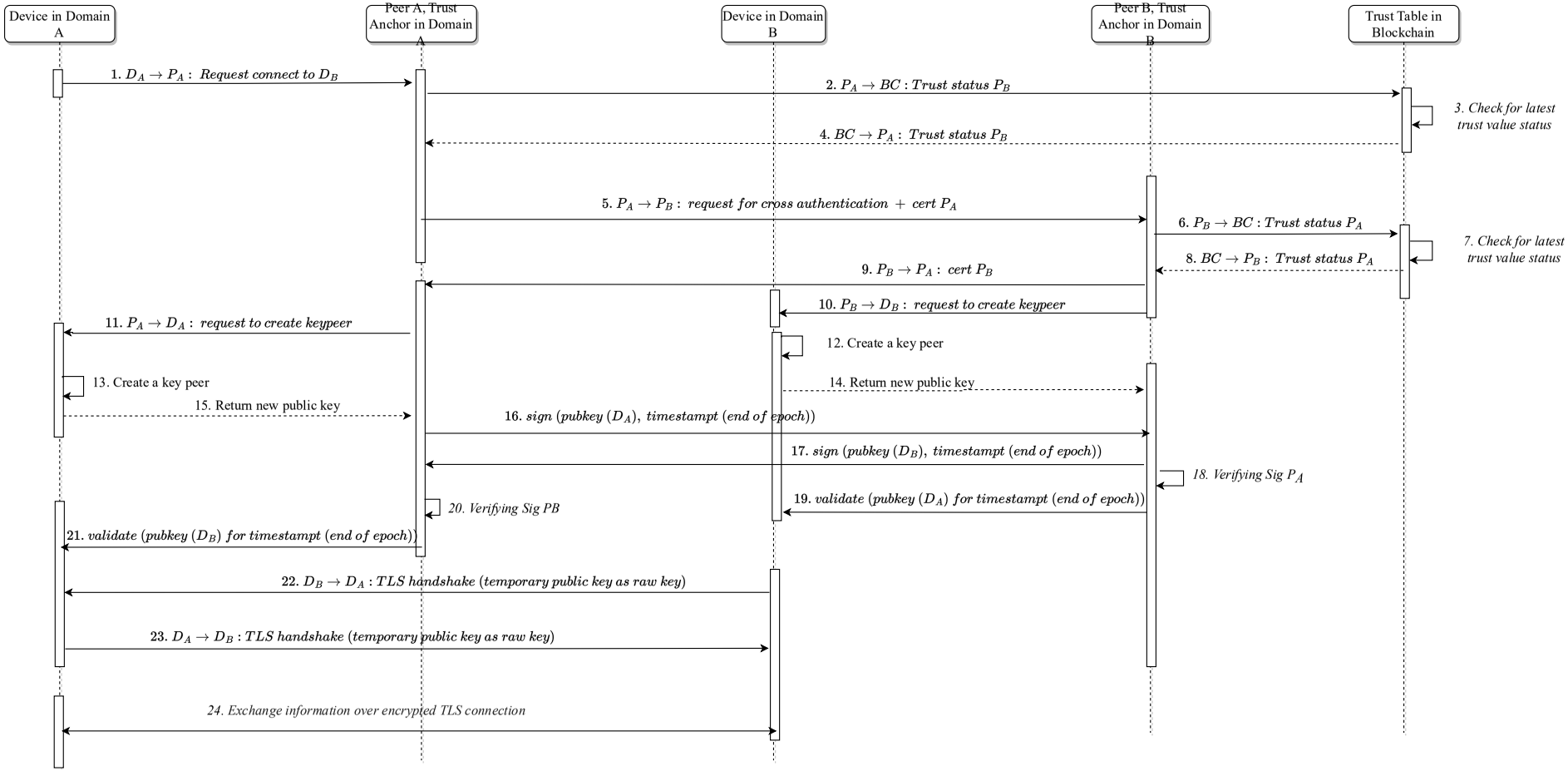


Figure 4.4: Cross Authentication Sequence Diagram.

4.3.4 Proof of Correctness for Cross-Domain Authentication

In this section, we formally prove the correctness and security guarantees provided by our cross-domain authentication model. Specifically, we demonstrate two critical security properties:

- **Forgery Resistance:** It is infeasible to produce a valid digital certificate without possessing the CA's private key.
- **Session Security Maintenance:** Once established securely, a cross-domain session remains secure and resilient to unauthorized interference.

We present these proofs using: (i) A reductionist argument demonstrating that forging a digital certificate contradicts the underlying digital signature scheme's security. (ii) An inductive argument proving that a secure session maintains its security after the initial authentication.

Throughout this proof, we explicitly refer to the "Schloss system", defined as our blockchain-supported TLS-based cross-domain authentication framework previously introduced in Sections 4.3.2 and 4.3.3.

Definition 5 (Digital Signature Scheme)**.** *(As defined previously in Chapter 2) We rely explicitly on a digital signature scheme Σ that is Existentially Unforgeable under Chosen-Message Attacks (EUF-CMA).*

Definition 6 (Digital Certificate for Cross-Domain Authentication)**.** *While digital certificates were introduced earlier (Section 4.3.3), here we explicitly highlight the incorporation of blockchain-managed trust values as a novel element: A digital certificate for an entity E is:*

$$Cert_E = (E, PK_E, T_E, \sigma_E),$$

where:

- *E and PK_E are previously defined (entity identity and public key),*
- *T_E is the newly introduced trust value assigned by the blockchain-based Trust Management System (TMS),*
- *$\sigma_E = \mathsf{Sign}(sk_{CA}, (E, PK_E, T_E))$ is a digital signature computed by a trusted CA over the tuple including trust values.*

Novelty: Incorporating blockchain-managed trust values into the certificate structure enhances decentralized verification capability and resilience.

Definition 7 (Authentication Function)**.** *Building on prior authentication definitions (4.3.3), we explicitly highlight the novel incorporation of the blockchain-derived trust threshold (τ) into the verification condition:*

$$Auth(Cert_A, Cert_B) = \begin{cases} 1, & \text{if } Verify(pk_{CA}, (E_A, PK_A, T_A), \sigma_A) = 1, \\ & \quad Verify(pk_{CA}, (E_B, PK_B, T_B), \sigma_B) = 1, \\ & \quad T_A, T_B \geq \tau, \\ 0, & \text{otherwise.} \end{cases}$$

Novelty: This definition explicitly incorporates a minimum trust threshold, reflecting real-time trust evaluations derived from blockchain-based trust management. This ensures flexibility and dynamic adaptation to changing trust conditions.

Definition 8 (Secure Cross-Domain Session)**.** *We define explicitly a secure cross-domain session as a session between authenticated entities D_A and D_B satisfying:*

- *Successful cross-domain authentication* ($Auth(Cert_A, Cert_B) = 1$).
- *Completion of a TLS handshake using ephemeral session keys derived from the authenticated certificates.*

Novelty: Clearly specifies the interplay between authentication and subsequent session establishment, emphasizing secure session continuity post-authentication.

Theorem 3: Unforgeability of Digital Certificates

Theorem 3. *Given that the digital signature scheme Σ is EUF-CMA secure, and assuming the blockchain-based Trust Management System (TMS) correctly computes and maintains trust values, no probabilistic polynomial-time adversary $\mathcal{A}$ can forge a valid digital certificate*

$$Cert^* = (E, PK, T, \sigma)$$

such that $Auth(Cert^*, Cert_B) = 1$ for a legitimate certificate $Cert_B$, except with negligible probability.

Proof. The proof follows by contradiction, similarly to classical EUF-CMA-based proofs, with the key novelty explicitly being the trust value component (T^*) managed by the blockchain-based TMS: Suppose, for contradiction, an adversary $\mathcal{A}$ produces a forged certificate

$$Cert^* = (E, PK, T, \sigma)$$

such that $\mathsf{Verify}(pk_{CA}, (E, PK, T), \sigma) = 1$ and the trust value T meets or exceeds the authentication threshold ($T \geq \tau$).

Such a forgery would imply $\mathcal{A}$ successfully generated a valid signature on a tuple including the blockchain-managed trust value, without access to the private signing key (sk_{CA}).

This directly contradicts the EUF-CMA assumption of the digital signature scheme Σ, as forging such a signature on arbitrary data (including trust values) is infeasible without the private key. Consequently, the existence of $\mathcal{A}$ violates the underlying assumption of the signature scheme's security, and thus such forgery can occur only with negligible probability. □

Theorem 4: Session Security Invariant

Theorem 4. *Let $\mathcal{S}(t)$ denote the state of an established cross-domain session at time t. If at time t_0, $Auth(Cert_A, Cert_B) = 1$ and the TLS handshake is successfully completed, then for all $t \geq t_0$, the session remains secure, provided that the underlying cryptographic primitives (TLS and digital signatures) remain secure.*

Proof. We prove the theorem by induction on the session lifetime. **Base Case:** At time t_0, the session is established after a successful TLS handshake following $Auth(Cert_A, Cert_B) = 1$. Hence, the session is secure at t_0. **Inductive Step:** Assume that at time t, the session remains secure; that is, all communications are encrypted and integrity is maintained by TLS. At time $t + 1$, any new message transmitted continues to be protected by the cryptographic guarantees of TLS. An adversary attempting to intercept, alter, or impersonate messages would need to break these guarantees. Since breaking TLS (or forging digital signatures) is computationally infeasible under our assumptions, the session remains secure at $t + 1$.

By induction, the secure session invariant holds for all $t \geq t_0$. □

Theorem 3 demonstrates that forging a valid digital certificate (and thereby bypassing cross-domain authentication) is computationally infeasible under the EUF-CMA security assumption of the digital signature scheme. Theorem 4 establishes that once a cross-domain session is securely established, its security persists throughout the session's lifetime. Together, these results provide a rigorous mathematical foundation for the security of the proposed cross-domain authentication scheme in the Schloss system.

4.3.4.1 Security Analysis

We conduct a security study of the proposed strategy, taking the aforementioned possible risks into consideration.

1. **The Man-in-the-Middle:** attack uses TLS session key encryption to symmetrically encrypt communication data between two parties, preventing data leakage and preventing attackers from deciphering future ciphertexts to obtain meaningful information, even if the data is stolen.[195].

2. **51% Attack:** The blockchain consensus method limits attackers' security to 51% of nodes or arithmetic power, making it unfeasible. To ensure authentication, peers verify the trust value of each subsystem, reducing the risk of connecting to a rogue node and ensuring a secure system [195].

3. **Replay Attack:** random values and a counter for nodes in each session are used to guarantee that communication messages remain current across sessions, avoiding replay attacks [195].

4. **Spam Attack:** Blockchain technology can protect against spam attacks by handling communication as transactions with a time stamp indicating a consensus phase. This prevents attackers from inserting spam messages, as they would be rejected by the consensus process [196].

The proposed schema provides a verification layer for a distributed system using blockchain authentication, enhancing security by enabling transparent communication between nodes. However, this mechanism can introduce new attack surfaces, necessitating a comprehensive security analysis to identify vulnerabilities. The analysis should evaluate the system's architecture, communication protocols, and access

controls, as well as potential attacks on data confidentiality, integrity, and availability. Implementing appropriate security measures and countermeasures can improve network security and ensure reliable communication between nodes.

4.4 Conclusion

This chapter has presented two contributions aimed at enhancing security aspects – identity management, authentication of IIoT environments. By addressing key gaps identified in existing solutions, these contributions provide foundational advancements essential for securing modern smart factories.

By combining digital wallet-based identity management with a blockchain-enabled cross-domain authentication framework, this chapter addresses two major hurdles in securing modern IIoT environments. Digital wallets decentralize and streamline identity handling, reducing single points of failure and administrative overhead. At the same time, cross-domain authentication ensures that trusted interactions can span multiple industrial networks, overcoming prior limitations on interoperability and scalability. Taken together, these two contributions form a more comprehensive solution for robust security and operational agility in next-generation smart factories.

Nevertheless, the cross-domain authentication framework also faces certain practical challenges:

- Performance and scalability management in blockchain networks, especially for extensive industrial deployments, may result in latency or throughput constraints.
- Maintaining resilience against emerging blockchain-specific vulnerabilities (e.g., consensus algorithm weaknesses and smart contract vulnerabilities) necessitates continuous monitoring and adaptation.
- Initial implementation complexity related to establishing decentralized trust networks among independent industrial stakeholders.

In conclusion, the contributions described in this chapter improve existing identity management and authentication paradigms in IIoT contexts. Despite certain operational complications, these contributions help to create the foundations for safe identity management and cross-domain authentication by addressing significant security holes. Building on these foundations, the following chapter will improve the entire security architecture by incorporating anomaly detection and evaluation tools. This connection is critical for proactively recognizing and responding to threats, resulting in full real-time security inside dynamic IIoT networks.

Chapter 5

Anomaly Detection and Anomaly Assessment in IIoT

Contents

Publications:

- Fatemeh Stodt, Fabrice Theoleyre, and Christoph Reich. "Advancing Network Survivability and Reliability: Integrating XAI-Enhanced Autoencoders and LDA for Effective Detection of Unknown Attacks". In: ***2024 20th International Conference on the Design of Reliable Communication Networks (DRCN), May 6-9, Montréal, Canada***. IEEE. 2024, pp. 9–16
- Fatemeh Stodt, Christoph Reich, and Fabrice Théoleyre (2025). "Context-Aware Anomaly Detection by Community Detection in the Internet of Things". In: ***Computer Communication***. Under review.

5.1 Introduction

Ensuring robust security within IIoT environments remains a critical challenge due to the highly dynamic, distributed, and heterogeneous nature of these systems. As IIoT networks become increasingly interconnected and exposed to sophisticated cyber threats, relying solely on identity management and static access control mechanisms is insufficient. Instead, proactive and adaptive security measures are needed to detect and assess anomalous behavior in real time.

Traditional anomaly detection approaches often depend on predefined signatures or simple behavioral baselines. While effective for known threats, such methods struggle with detecting emerging or context-specific anomalies, including zero-day attacks and subtle internal deviations. Furthermore, the complex interaction patterns within IIoT systems frequently lead to high false-positive rates when conventional techniques are applied.

This chapter introduces two complementary contributions that aim to address these challenges by designing anomaly detection methods tailored to the specific needs of IIoT environments:

- **AE-LDA Hybrid Anomaly Detection** (Section 5.3): This method combines Autoencoders (AEs) for unsupervised feature extraction with Linear Discriminant Analysis (LDA) for statistical classification. Unlike earlier approaches that employ isolated machine learning models [94], AE-LDA offers enhanced detection accuracy by capturing nonlinear data patterns and applying discriminative class boundaries. It is particularly effective at identifying novel and previously unseen attacks while maintaining low computational complexity.

- **Context-Aware Behavioral Anomaly Detection** (Section 5.4): While AE-LDA provides strong detection performance, it lacks contextual awareness. This second contribution addresses that gap by modeling device interactions as dynamic communities and analyzing temporal and structural changes in communication patterns. By incorporating context (such as timing, expected communication flows, and operational roles) this approach improves interpretability and reduces false positives, offering a fine-grained, behavioral anomaly detection mechanism suited to the evolving nature of IIoT systems.

Together, these contributions form a comprehensive framework for anomaly detection in IIoT networks. By integrating deep learning with graph-based context modeling, the proposed methods enable real-time, scalable, and context-sensitive security monitoring, aligning with the operational demands of modern industrial systems.

Addresses Research Questions:

- **How can hybrid and context-aware anomaly detection methods improve the real-time identification and assessment of sophisticated security threats within dynamic IIoT networks?**

5.2 Problem Statement

IIoT networks are inherently complex due to their heterogeneous components, distributed architecture, and the continuous exchange of large volumes of data. This complexity significantly increases the attack surface, making IIoT systems vulnerable to various security threats, both internal and external.

Ensuring the security of IIoT networks remains a significant challenge due to their inherent complexity, heterogeneous nature, and continuous exposure to evolving internal and external threats [197, 198]. Traditional anomaly detection techniques, such as signature-based methods, rely heavily on predefined patterns and thus cannot effectively detect unknown or sophisticated threats, including zero-day attacks [199]. Consequently, methods leveraging unsupervised deep learning, particularly autoencoder-based anomaly detection approaches, have gained attention due to their ability to identify deviations from normal behaviors without relying on labeled attack signatures [200, 201].

Nevertheless, current autoencoder-based anomaly detection techniques exhibit critical limitations. Approaches combining autoencoders with One-Class Support Vector Machines (OC-SVM) [98] demonstrate promising results, but their significant computational complexity limits real-time applicability. Moreover, the performance of OC-SVM deteriorates in high-dimensional IIoT datasets, often causing high false-positive rates due to its dependence on a single-class data distribution.

Other methods, such as Memory-Augmented Autoencoders (MemAE) [94], attempt to mitigate autoencoder overgeneralization through external memory modules. However, these approaches introduce substantial memory overhead, making them impractical for resource-constrained environments typical of industrial settings. Additionally, memory-based methods may still overlook context-specific anomalies due to insufficient consideration of contextual features.

Hence, an effective anomaly detection solution for IIoT systems must address the following key challenges:

- Achieving computational efficiency suitable for real-time detection.
- Managing dimensionality effectively to handle high-dimensional IIoT data streams.
- Maintaining low false-positive rates while enhancing the ability to detect context-specific anomalies.

5.2.1 Context-Aware Anomaly Detection

IIoT networks typically consist of interconnected sensors, actuators, gateways, machines, and centralized monitoring systems, all operating under specific constraints and operational policies (see Figure 5.1). Due to the tight integration of these networks with physical processes, even minor false alarms can severely disrupt critical industrial workflows. Conventional anomaly detection methods frequently struggle to differentiate between harmless operational variations (such as scheduled production adjustments) and genuine security threats like malicious cyber-attacks.

In this work, the term **context** specifically refers to supplementary information related to the operational state and environmental conditions of the IIoT system, which helps interpret and evaluate the significance of observed behaviors. More concretely, *contextual information* includes attributes such as:

- **Temporal Context:** Operational schedules, time of day, periodic tasks, and seasonal changes.
- **Communication Context:** Typical communication patterns, expected sender-receiver pairs, message frequencies, and standard protocols.
- **System State Context:** Machine operational status, maintenance schedules, known operational modes, or production phases.

Context-aware anomaly detection methods incorporate these contextual dimensions to determine whether a detected deviation genuinely represents a security threat or simply reflects normal system behavior under given circumstances. By explicitly integrating such context, anomaly detection systems significantly reduce false-positive rates and improve their capability to recognize subtle, context-specific security threats [202].

Throughout this thesis, the term "context-aware" refers specifically to approaches that utilize these additional dimensions of information beyond basic statistical or signature-based metrics to enhance the accuracy of anomaly detection in IIoT environments.

5.2.1.1 Types of Anomalies

Our proposed context-aware system detects two main categories of anomalies:

Attacks: Malicious entities attempt to infiltrate the system, either to steal data or to compromise safety. For example, an attacker might force a machine to operate outside its safety zone in a connected smart factory [203].

Misbehavior: The IIoT infrastructure exhibits unexpected behavior that does not necessarily stem from a deliberate attack. Within a network, such misbehavior might involve:

- **Temporal Anomalies:** Communication occurring at unusual times (e.g., after work hours).
- **Behavioral Anomalies:** New or unexpected interactions between devices.
- **Statistical Anomalies:** Significant deviations in metrics such as packet sizes or flow duration.

In a typical IIoT network, sensor data flows from edge devices to a central Monitoring System, while machines exchange operational commands within predefined workflows. For instance, a packaging machine might only communicate with its sensors or the Monitoring System during production hours. Any communication outside of these patterns could indicate an anomaly.

Figure 5.1 provides a high-level view of a generic IIoT network, illustrating how various devices and systems interconnect to support industrial operations.

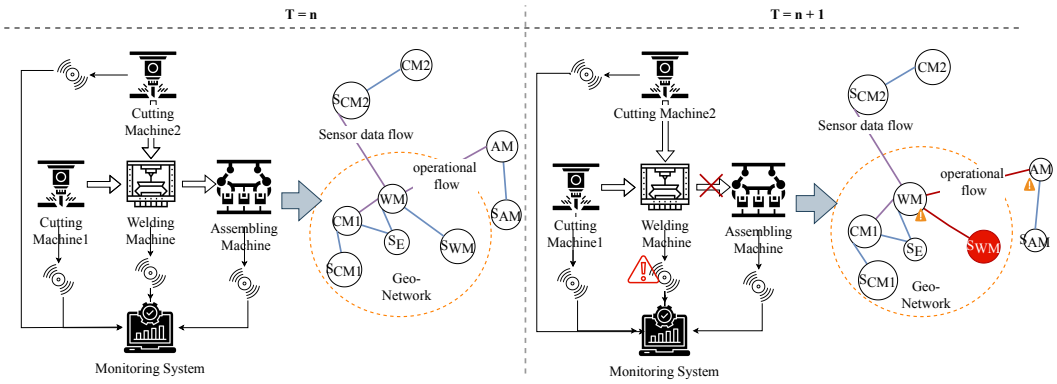


Figure 5.1: Generic IIoT Network and Attack Scenarios.

5.2.1.2 Example Anomaly Scenarios

Scenario 1: Unauthorized Device Access An attacker introduces a rogue device that impersonates a legitimate node, attempting unauthorized actions such as:

- **Unexpected Communication Patterns:** Sending commands to machines that it is not authorized to control.
- **Violation of Temporal Policies:** Attempting to access the network or critical systems during non-production hours.

These deviations from established communication and time-based policies disrupt the contextual integrity of the network. A context-aware detection system flags such anomalies by comparing observed activities against known legitimate usage patterns.

Scenario 2: Misconfigured or Compromised Device Behavior A legitimate device (e.g., a machine or sensor) becomes compromised or is misconfigured, leading to:

- **Structural Anomalies:** Interacting with devices it does not typically communicate with (e.g., a packaging machine suddenly contacting a finance server).
- **Temporal/Statistical Anomalies:** Sending a high volume of data or operating during non-production hours.

Such deviations break known contextual norms for device behavior and can be automatically flagged for further investigation.

5.2.1.3 Why Context Matters

Contextual data such as when (time), how (protocol, volume), and with whom (source/destination) a device communicates enables the anomaly detection system to:

- **Reduce False Alarms:** Legitimate production changes or maintenance activities that appear unusual to a purely pattern-based system can be validated through contextual checks (e.g., planned overnight maintenance).
- **Improve Detection Accuracy:** Threats become more apparent when they deviate from real-world operational constraints (e.g., an unauthorized machine controlling manufacturing equipment).
- **Preserve Operational Continuity:** By focusing on genuine threats, the system avoids shutting down or raising alarms for benign operational shifts.

By integrating rich contextual insights into the detection process, context-aware anomaly detection ensures a more adaptive and precise security mechanism that distinguishes legitimate operational variations from true security threats. Graph Neural Networks (GNNs) can further enhance detection by modeling the IIoT network as a communication graph where nodes represent devices and edges represent interactions. Because GNNs can learn structural, temporal, and statistical relationships in these graphs, they can identify nodes or edges that deviate from expected patterns enabling the system to flag malicious activities such as malware propagation or unauthorized access attempts.

5.2.2 Limitations Motivating GNN-Based Approach

GNNs have recently gained popularity in anomaly detection due to their inherent capability of analyzing network topologies. However, current GNN-based anomaly detection methods exhibit critical limitations that restrict their practical deployment in dynamic IIoT environments.

For instance, E-GraphSAGE [111] extends node dependencies to capture distant network interactions but does so at the expense of significantly increased computational complexity and response delays, hindering real-time anomaly detection capabilities required in rapidly changing industrial scenarios.

Similarly, Altaf et al. [113] further developed the E-GraphSAGE model by introducing sophisticated but memory-intensive sampling techniques, resulting in substantial resource consumption that makes such approaches unsuitable for large-scale, resource-constrained IIoT deployments.

Temporal Graph Networks (TGNs) proposed by Rossi et al. [117] address the evolving nature of graphs effectively; however, they are primarily designed and optimized for social networks, lacking straightforward applicability and efficient adaptation mechanisms needed specifically for industrial IoT contexts.

These limitations directly motivate the necessity of developing a context-aware, efficient GNN-based anomaly detection approach specifically tailored for IIoT networks. In contrast, our proposed Context-Aware Behavioral Anomaly Detection method leverages community structure analysis and explicitly incorporates context-aware features. This enables efficient, real-time detection of context-driven anomalies, significantly improving accuracy, scalability, and resource efficiency compared to existing state-of-the-art methods.

5.3 AE-LDA Hybrid Anomaly Detection

5.3.1 Design Goals and Assumptions

Our framework is strategically designed to provide a comprehensive and resilient approach to network intrusion detection, addressing the following key goals and assumptions:

- **Enhancing Network Reliability With Real-Time Monitoring:** System designed for network reliability by incorporating real-time monitoring capabilities. This approach manages continuous surveillance and intrusion detection activities efficiently, minimizing performance impact. By providing a security solution that operates within real-time constraints, we safeguard against intrusions and maintain network stability. This real-time aspect enables immediate detection and response to potential threats, ensuring uninterrupted and stable network functioning.
- **Using Explainable AI (XAI) for Transparent Decision-Making for selecting Features:** System incorporates XAI to enhance transparency and understanding in decision-making, enabling network administrators to better understand system alerts, thereby promoting informed and effective security management.
- **Detection of Unknown and Evolving Threats:** System focuses on identifying and mitigating non-conventional network threats, such as zero-day attacks and novel malware types, by analyzing deviant network patterns, providing a robust defense against emerging security risks.
- **Resilience Against Evasion Techniques:** Recognizing the evolving nature of cyber threats, our system is designed to remain effective against sophisticated evasion tactics employed by attackers. This involves maintaining high detection accuracy even as attackers modify their strategies to evade traditional security measures.

Our design approach aligns with key areas to create a robust, adaptable, and efficient framework for network intrusion detection, addressing current security challenges and anticipating future threats for long-term network system resilience and reliability.

Our proposed approach for anomaly detection, illustrated in Fig. 5.4, aims to improve both the accuracy and interpretability of network security anomaly detection. It consists of two core components: a feature extraction phase that preprocesses and transforms raw network data into structured representations, and an anomaly detection model that combines Autoencoders (AE) with Latent Dirichlet Allocation (LDA) to capture both structural deviations and semantic patterns indicative of abnormal behavior.

5.3.1.1 Feature Extraction

The preprocessing process involves extracting significant features from raw Packet Capture (PCAP) files to condense network interactions into clear patterns and adapt

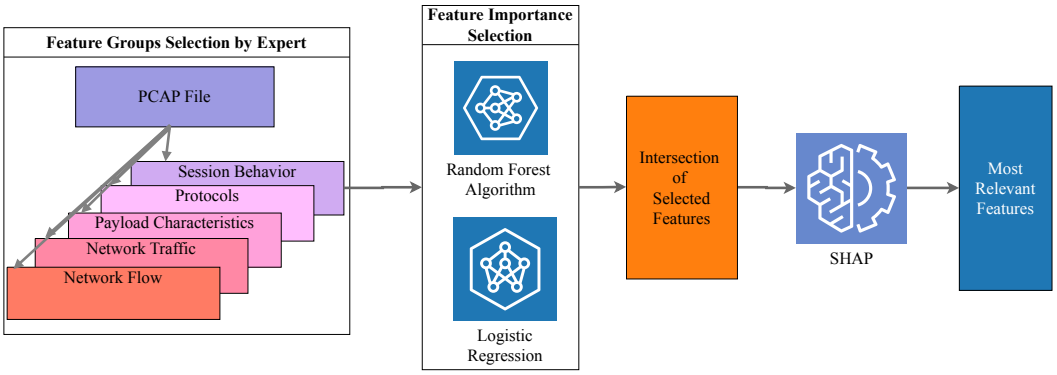


Figure 5.2: The Process Workflow of Feature Selection.

the data for machine learning model training (see Fig 5.2). PCAP files provide a holistic view of network traffic, allowing insights into network behavior, vulnerabilities, and malicious activities. The main challenge is to methodically identify and categorize the crucial data within these packets. The approach focuses on devising an efficient set of features for further analysis. The extracted features are divided into five categories: Network Traffic Features, Session-related Information Features, Network Flow Features, Protocol-specific Features, and Payload Characteristics Features. This strategy lays the groundwork for a comprehensive network traffic study.

Our feature extraction process combines meticulously the robustness of a Random Forest algorithm with the detailed insights of SHapley Additive exPlanations (SHAP) [204]. The SHAP value determines how a given feature explains (impacts) the model's prediction. We initiate with a Random Forest to identify key features, where the Gini importance of each feature f is calculated as:

$$\text{Gini Importance}(f) = \frac{1}{N} \sum_{i=1}^{N} \text{Impurity Decrease}_i(f) \tag{5.1}$$

Where N is the number of trees, and $\text{Impurity Decrease}_i(f)$ represents the decrease in impurity in the i-th tree due to feature f.

To further refine and understand the importance of these features, we employ SHAP values. For a feature f, its SHAP value is determined by:

$$\text{SHAP}(f) = \frac{1}{M} \sum_{j=1}^{M} \text{Marginal Contribution}_j(f) \tag{5.2}$$

Where M is the number of all possible permutations of features, and $\text{Marginal Contribution}_j(f)$ denotes the change in the prediction outcome when including feature f in the j-th permutation.

This combined approach of using Random Forest feature importance alongside SHAP-based explanations ensures that our selected features are both statistically robust and clearly interpretable. The choice of Random Forest for feature selection is justified by its proven stability and resilience against noise, as shown by previous studies [205, 206]. Moreover, the integration of SHAP values provides de-

Algorithm 4: Feature Extraction Algorithm

Input: Dataset, Random Forest model, importance threshold
Output: Reduced feature set and SHAP values for interpretation

Train a Random Forest model on the dataset;
foreach ***feature*** f ***in the dataset*** **do**
    Calculate Gini importance of f using eq. (5.1);
Prune features based on the importance threshold to reduce model complexity;
foreach ***remaining feature*** f **do**
    Compute SHAP values for f using eq. (5.2) to understand its contribution;
Utilize optimized TreeSHAP for large datasets to balance detail and efficiency;

tailed insight into how each feature impacts model predictions, enhancing model transparency and interpretability [204, 207].

Compared to traditional feature selection methods relying solely on correlation analysis or univariate statistical tests, our combined strategy systematically addresses both predictive power and interpretability. Specifically, it balances model complexity and comprehensibility, enabling the construction of more reliable anomaly detection models. Consequently, this method establishes a stronger methodological foundation, as it directly contributes to improved predictive accuracy and clarity in decision-making processes – both crucial for effective anomaly detection in sensitive industrial environments [207].

5.3.2 Anomaly Detection Model (AE-LDA)

Then, we have to detect anomalies in the traffic. We rely on an autoencoder with LDA to characterize the usual network traffic, and thus, to detect anomalies.

Autoencoder for Anomaly Detection We train the autoencoder to capture the normal behavior of network traffic. Thus, we train the model with all the data which is i) generated by the network when we are sure that no attack occurs (*i.e.*, at the first stage of the deployment), ii) a training dataset without data labeled with an attack. Very classically, the training objective of the autoencoder is to minimize the Mean Squared Error (MSE) between the input vector x and its reconstruction $\hat{x}$, given by:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2 \tag{5.3}$$

where n is the number of features. Anomalies are identified when the reconstruction error exceeds a predefined threshold θ.

The specific structure of the autoencoder is illustrated in Fig. 5.3. Its architecture was carefully selected to capture the intricate patterns effectively inherent in the network traffic data. The input layer comprises neurons corresponding to the 17 selected features, derived from the earlier feature-selection process.

The encoding part systematically reduces the input dimensionality to efficiently identify essential traffic characteristics. Through a hyperparameter optimization process using Optuna [208], we determined that employing two hidden encoding layers one with 8 neurons and a subsequent layer with 4 neurons provided an optimal balance between model complexity and representation capability. We adopted

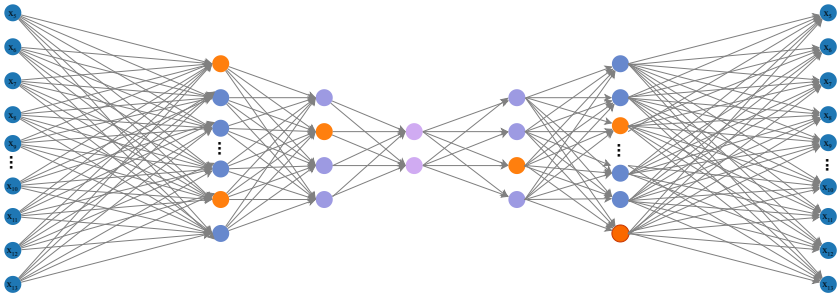


Figure 5.3: The Autoencoder Structure.

the Rectified Linear Unit (ReLU) activation function due to its ability to mitigate the vanishing gradient problem and improve convergence speed, aligning with findings from previous deep learning studies [209]. To enhance model generalization and reduce the risk of overfitting, we incorporated dropout layers with a dropout rate of 30%, which was selected based on experimental validation and hyperparameter optimization results. Convolutional or deconvolutional layers were deliberately omitted, given that network traffic features lack the spatial dependence characteristic of image data, making convolutional methods less suitable for this scenario. The bottleneck (latent space) of the autoencoder consists of two neurons, forming a highly compressed representation that efficiently captures core traffic patterns critical for anomaly detection. The decoding layers mirror the encoding structure, systematically reconstructing the input data. The output layer utilizes a sigmoid activation function to match the scale and bounded nature of the original input features. For training, an Adam optimizer with a learning rate of 0.001 and MSE loss function was used, selected based on experimental performance. An early stopping strategy was applied to further prevent overfitting, limiting training to a maximum of 50 epochs with shuffled mini-batches of size 256.

LDA for Attack Classification Our primary objective is not only detecting anomalous network traffic but also effectively classifying detected anomalies into specific threat categories. Such classification capability is crucial, as it provides security administrators actionable insights, facilitating targeted and efficient security interventions. Merely identifying anomalies without categorization limits the practical effectiveness of the detection system, since appropriate countermeasures often depend on the type of attack.

We selected LDA as our classification approach for several key reasons. First, LDA inherently provides interpretable results due to its linear decision boundaries, aligning with our overarching goal of achieving explainability in anomaly detection [207, 210]. Second, previous studies demonstrate that LDA maintains robust classification performance even when training data is limited or imbalanced, conditions frequently encountered in cybersecurity contexts [211]. Finally, empirical results from our preliminary experiments confirmed that LDA offers an effective balance between computational simplicity and classification accuracy, especially when integrated with autoencoder-based anomaly detection. In particular, recent literature emphasizes that LDA tends to require significantly lower computational resources compared to alternative classifiers like Support Vector Machine (SVM) or

Random Forest, while still maintaining competitive accuracy on structured tabular datasets [212, 211].

LDA operates by identifying linear combinations of features that optimally separate predefined classes. In our scenario, we explicitly construct these classes from labeled datasets representing normal network traffic as well as known attack types, such as "Attack X," "Attack Y," and so forth. Thus, after detecting anomalies using the autoencoder (trained on normal traffic data), the LDA model classifies these anomalies into known threat categories.

This hybrid AE-LDA methodology specifically addresses a key limitation of autoencoder-based detection: although autoencoders effectively detect novel anomalies (e.g., zero-day attacks), they cannot inherently classify anomalies into known attack types. Conversely, LDA cannot identify entirely unknown threats without labeled training data but excels at differentiating among known categories. Thus, combining these two complementary approaches achieves robust anomaly detection and clear threat classification simultaneously.

Mathematically, the LDA classification decision is based on Bayes' rule, which calculates the posterior probability of an anomaly belonging to a specific category given its observed features. The decision rule used by LDA is:

$$\text{Decision Rule} = \log \frac{P(y|x)}{P(\neg y|x)} \tag{5.4}$$

where $P(y|x)$ represents the posterior probability that the detected anomaly belongs to a specific attack class y, given the feature vector x.

The integration of anomaly detection and classification in our AE-LDA framework is summarized clearly in Algorithm 5.

Algorithm 5: AE-LDA Algorithm

Input: Trained autoencoder on normal network traffic data, new data points, anomaly threshold θ

Output: Flagged anomalies and their categories

```
foreach new data point do
    Calculate the reconstruction error;
    if error exceeds θ then
        Flag the data point as an anomaly;
        Use LDA to classify the anomaly into a specific category;
```

The overall workflow for anomaly detection and classification using AE-LDA is illustrated in Fig. 5.4.

5.3.3 Experimental Evaluation

The proposed approach's performance is evaluated across various dimensions to validate its effectiveness in detecting and classifying network anomalies, focusing on accuracy and real-time responsiveness. We exploit two different datasets (*i.e.*, CICIDS2017 [213], Kitsune [104]) to illustrate the genericity of our approach and its robustness to detect anomalies in a wide range of network activities and attack

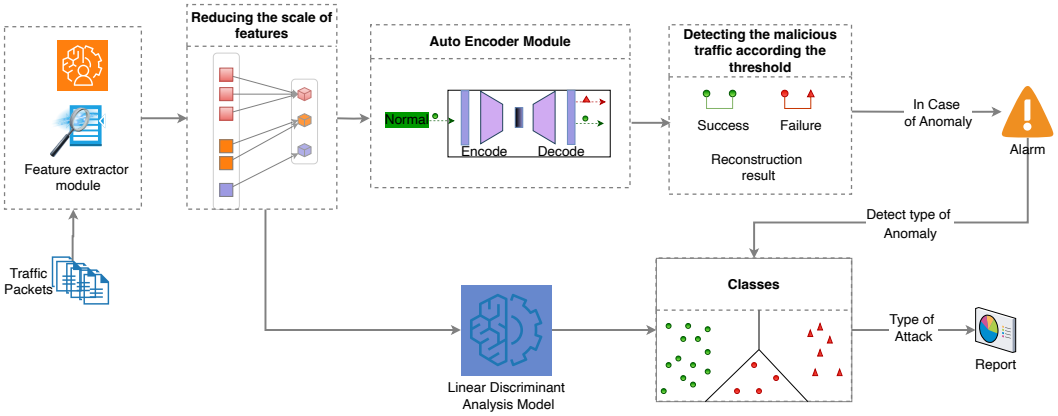


Figure 5.4: The Process Workflow of Anomaly Detection.

scenarios. We measure usual key performance metrics expressed as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.5}$$

$$Precision = \frac{TP}{TP + FP} \tag{5.6}$$

$$Recall = \frac{TP}{TP + FN} \tag{5.7}$$

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{5.8}$$

$$Prediction\ time = \frac{T_{\text{total}}}{N_{\text{packets}}/N_{\text{packets per flows}}} \tag{5.9}$$

Where TP, TN, FP, and FN stand for respectively True Positives, True Negatives, False Positives and False Negatives. Furthermore, accuracy measures the proportion of correct predictions (both true positives and true negatives) out of all predictions made by a model.

5.3.3.1 Preliminary Performance Evaluation

The study evaluates a model's ability to detect anomalies and unknown attacks, using benign traffic only for training. The model's proficiency in detecting unknown attacks is crucial for real-world applications, with an F1-score of 0.9417 and an accuracy of 0.9590 observed in experiments. This demonstrates the model's robustness and adaptability to evolving security challenges. The model's efficacy in recognizing zero-day attacks underpins advanced anomaly detection techniques. The Receiver Operating Characteristic (ROC) curve shows (see Fig. 5.5) the balance between sensitivity and specificity across operational thresholds, indicating consistent performance under different conditions.

5.3.3.2 Performance on CICIDS2017 Dataset

We compare our proposed AE-LDA approach against three baseline models frequently cited in anomaly detection literature, specifically chosen for their representative capabilities and relevance to our scenario:

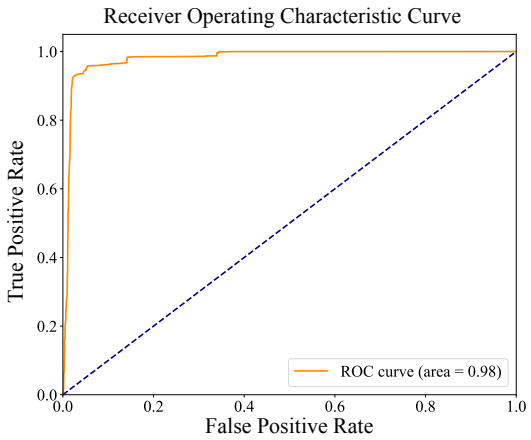


Figure 5.5: ROC Curve Depicting the Model's Sensitivity and Specificity Across Varying Thresholds.

- **One-Class SVM (OCSVM)** [91] serves as a classical baseline, widely used in cybersecurity due to its simplicity, interpretability, and established performance for anomaly detection scenarios [199].
- **Autoencoder combined with One-Class SVM (AE)** [98] represents recent advancements by leveraging deep learning (autoencoder) for feature extraction combined with the robust classification capabilities of a one-class SVM in the latent space. This method is an established benchmark demonstrating improved performance over classical anomaly detection methods.
- **Memory-Augmented Autoencoder (MemAE)** [94] further advances deep learning-based anomaly detection by integrating a memory module to prevent the autoencoder from overly generalizing and thereby achieving superior detection performance over traditional autoencoder-based models.

These methods collectively cover classical machine learning techniques, standard deep learning approaches, and state-of-the-art memory-augmented deep learning methods, providing a thorough comparative perspective against which to evaluate our AE-LDA model.

We evaluate the models using complementary metrics to provide a holistic performance evaluation:

- **Area Under the ROC Curve (AUROC)** assesses overall discriminative power in differentiating normal traffic from attacks. AUROC is widely regarded as an essential measure for anomaly detection since it effectively captures the trade-off between true positive rate and false positive rate [214].
- **MSE** measures the accuracy of data reconstruction by the autoencoder, reflecting the model's fidelity in representing normal traffic patterns. A lower MSE indicates higher accuracy in learning normal behavior patterns and thus improved anomaly detection performance.

- **Detection Time (latency)** is crucial for practical deployment in real-time Intrusion Detection System (IDS), as it directly influences system responsiveness and the ability to mitigate security threats immediately upon detection.

Together, these metrics ensure that our evaluation covers accuracy (AUROC), quality of learned representations (MSE), and practical viability in terms of speed (detection latency).

Table 5.1 and Figure 5.6 present the comparative AUROC results for all evaluated models, demonstrating the effectiveness of our AE-LDA approach.

Table 5.1: Comparison of AUC Performance for CICIDS2017 for Different Models.

Model	AUROC
OCSVM [91]	0.7684
AE [98]	0.8758
MemAE [94]	0.9101
AE-LDA (Our Model)	**0.9800**

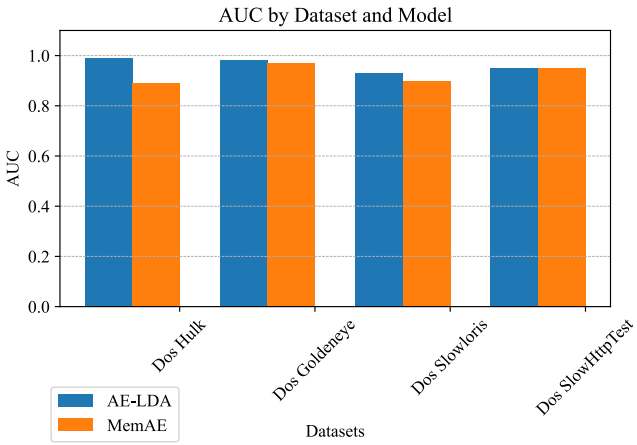


Figure 5.6: Comparative ROC Curve Analysis on CICIDS2017 Dataset.

Table 5.2 provides detailed metrics of AE-LDA across specific DoS attacks from CICIDS2017, clearly demonstrating robust and consistent performance in terms of accuracy, reconstruction error threshold, MSE, AUROC, and notably low detection latency (<12 ms).

Table 5.2: Detailed Performance Metrics of AE-LDA on CICIDS2017 Attacks.

DoS Attacks	Accuracy	Reconstr. Error Threshold	MSE	AUC	Detection Time (ms)
Hulk	0.9811	31.1040	4.1459	0.99	11.99
Goldeneye	0.9800	31.0997	1.0230	0.9772	11.93
Slowloris	0.9800	31.2328	0.8357	0.93	11.77
SlowHttpTest	0.9873	31.1841	0.7761	0.95	12.04

As evidenced by the results, AE-LDA significantly outperforms competing models, including MemAE, across key performance metrics. Its superior AUROC performance (0.98 vs. 0.91 for MemAE) demonstrates better discriminative capability. Additionally, AE-LDA maintains consistently low detection latency, ensuring practical real-time deployment. Such robust performance sets a new benchmark in IDS research, providing clear motivation for further development of integrated autoencoder-classifier frameworks in cybersecurity.

It is worth noting that Griffin [104] didn't provide their implementation. Thus, we are not able to compare AE-LDA with Griffin using the CICIDS2017 dataset.

We provide full access to our code [215] on GitHub to ensure that other researchers and practitioners can validate, reproduce, and build upon our work. This transparency is, to our mind, crucial in the field of cybersecurity.

5.3.3.3 Performance on Kitsune Dataset

We compare AE-LDA with Griffin [105] and the approaches already included in the original paper using the Kitsune dataset [104]. Since the code of Griffin is not available online, we can only compare our solution with Griffin using the same dataset, extracting their original results directly from their paper.

As depicted in Fig. 5.7 and Tab. 5.3, our model, incorporating an autoencoder and LDA, exhibits a strong average detection capability across various network scenarios. Griffin outperforms AE-LDA only for the detection of the SSD Flood attack, which is also well detected by other competitors.

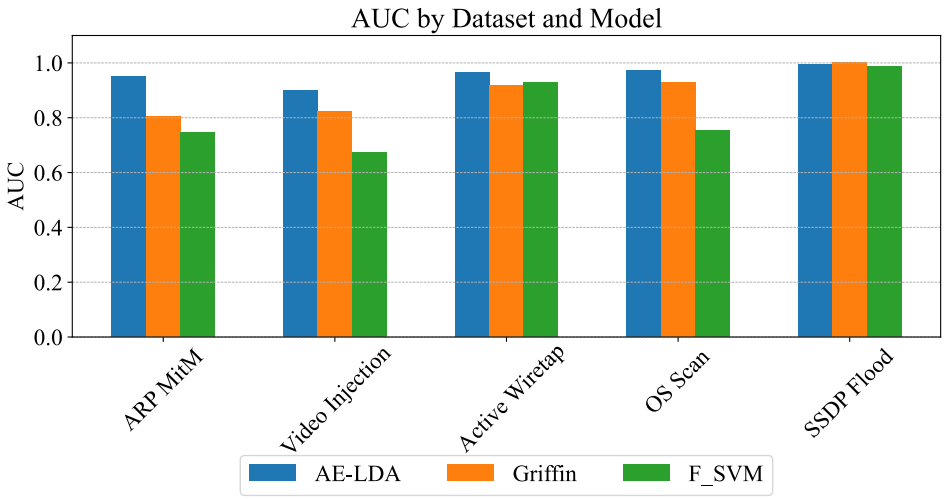


Figure 5.7: Performance Evaluation on the Kitsune Dataset.

The model integrates LDA with autoencoder, demonstrating high AUROC values for classifying network anomalies. It maintains consistent performance across various attack scenarios and is openly available, proving its practical applicability and reliability in real-world security contexts, highlighting its practical applicability and reliability.

Table 5.3: Detection Accuracy Comparison on Kitsune Dataset.

Method	AE-LDA	Griffin	pcStream2	F_SVM	F_RF
ARP MitM	0.9487	0.8048	0.7219	0.7452	0.6512
Video Injection	0.9007	0.8237	0.5816	0.6718	0.6139
Active Wiretap	0.9669	0.9188	0.7413	0.9281	0.7634
OS Scan	0.9713	0.9281	0.7513	0.7517	0.7212
SSDP Flood	0.9945	0.9999	0.9971	0.9876	0.8674

5.4 Considering Context

Our primary objective in proposing a community-based approach is to effectively detect anomalies in industrial networks by analyzing communication patterns among devices. Industrial systems typically exhibit strict and predictable communication structures, where only limited device interactions are permissible. Identifying deviations from these patterns is crucial for promptly detecting potential threats, intrusions, or operational anomalies. Traditional anomaly detection methods often ignore the relational and contextual aspects of communication patterns, leading to increased false positives or missed threats.

To address this gap, our approach leverages community detection techniques within multi-edge graphs, explicitly integrating contextual information about device interactions. Such a graph-based approach facilitates detecting clusters of closely interacting devices and allows monitoring their evolution over time. Changes in these communication communities can signal anomalous or malicious behavior, thus enhancing detection precision.

We specifically adopt a multi-edge graph approach for community detection because it naturally captures complex relationships among industrial network endpoints. Each edge type corresponds to a different dimension of interaction between devices, enabling the representation of multi-faceted relationships, such as temporal patterns, behavioral interactions, and policy constraints.

Importantly, we incorporate contextual information, defined as attributes describing the operational environment, timing, or situational circumstances influencing the behavior of network interactions. Contextual integration is essential since normal communication patterns in industrial environments heavily depend on factors like production schedules, expected device roles, and predefined policies. By explicitly modeling these contextual factors, our approach significantly improves anomaly detection accuracy and reduces false alarms.

Our proposed multi-edge graph considers the following specific feature types:

- **Network Communication Features:** These include direct metrics from packet flows, protocols, and header-level attributes (e.g., packet size, inter-arrival times). These features reflect fundamental communication patterns crucial for baseline traffic behavior identification.
- **Contextual Features:** Attributes capturing temporal and behavioral dynamics, such as operational schedules, interaction timing, or typical device communication sequences. These features allow the detection algorithm to distinguish routine operational variations from truly anomalous activities.

- **Knowledge-Based Features:** These encode expected norms, logical constraints, and policy-driven rules governing permissible communications between devices. Examples include:
  - *Packet Size Averages (Pkt Size Avg, Fwd Seg Size Avg, Bwd Seg Size Avg)*, which represent expected typical data transfer volumes.
  - *Flow Flags (e.g., SYN, ACK)*, representing expected protocol behaviors aligned with operational rules and device roles.

We aggregate these features at both the packet and flow levels. Flow-level aggregation provides higher-level insights into interaction patterns, effectively summarizing behaviors over broader communication sequences. Features like inter-arrival times and protocol flags specifically enhance our capacity to identify anomalous behaviors indicative of security threats, including unexpected traffic surges or misuse of network protocols.

By combining graph-based community detection with contextual and knowledge-driven insights, our method provides a robust anomaly detection approach tailored specifically to industrial network constraints, significantly surpassing traditional methods reliant solely on statistical deviations from baseline traffic.

5.4.1 Context-Aware Community-Based Multi-Graph Anomaly Detection

This section introduces our pipeline for anomaly detection, integrating multi-edge graph construction, community detection, and a HeteroGNN [216].

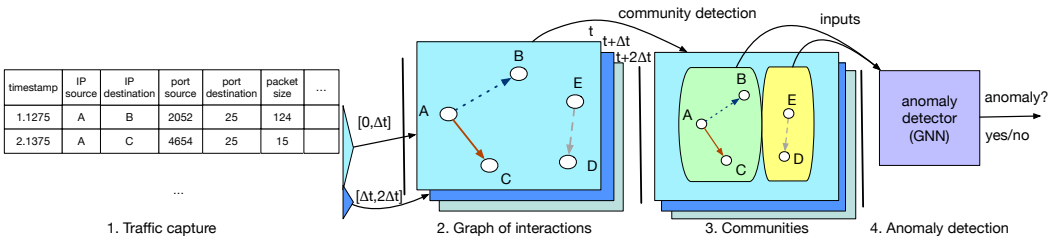


Figure 5.8: Steps of Our Community-Based Anomaly Detection Approach.

5.4.1.1 Big Picture and Steps Orchestration

The pipeline, illustrated in Figure 5.8, combines temporal, contextual, and structural data to detect anomalies in dynamic network environments. Algorithm 6 provides an overarching view of how these components interact.

The pipeline begins with network traffic data collection, followed by preprocessing to retain relevant features. The preprocessed data is used to construct a time-evolving multi-edge graph $G_t = (V_t, E_t)$, where V_t represents the nodes (IP entities) and E_t denotes edges capturing multiple interaction types. Each edge type (communication, context, knowledge) is annotated with weights computed from normalized features, ensuring a detailed representation of network interactions. Communities

are then detected within G_t to identify clusters of nodes with similar patterns, providing contextual insights into network behavior.

The enriched graph, with annotated nodes and edges, is processed using a HeteroGNN for edge classification, flagging anomalous edges. The pipeline operates in a sliding window framework $[t - W, t]$, ensuring real-time updates as new data arrives.

Algorithm 6: Sliding Window Orchestration

Input: W: window size, Δt: frequency of updates, HeteroGNN model (trained or partially trained), θ: anomaly threshold
Output: Anomalous edges flagged over time

Initialize $\mathcal{B}_0 \leftarrow \emptyset$; `// Initial flow buffer`
$t \leftarrow 0$;
while ***True*** **do**
  Wait until $t + \Delta t$;
  $t \leftarrow t + \Delta t$;
  Update buffer: Remove flows older than $t - W$, add new arrivals Δ_t;
  Build $G_t(V_t, E_t)$ using Algorithm 7;
  Run label propagation: Compute community labels π_t;
  foreach $v \in V_t$ **do**
    $X_t(v)$.community_label $\leftarrow \pi_t(v)$;
  Convert (V_t, E_t, X_t) to `HeteroData` structure;
  Perform inference with HeteroGNN: $Z_t \leftarrow \text{HeteroGNN}(X_t, E_t)$;
  Detect anomalous edges:;
  foreach $(u, v, k) \in E_t$ **do**
    $p_{\text{attack}} \leftarrow \text{Softmax}(\text{EdgeClassify}(Z_t(u), Z_t(v)))[1]$;
    if $p_{\boldsymbol{attack}} > \theta$ **then**
      Flag edge (u, v, k) as anomalous;

5.4.1.2 Time-Dependent Multi-Graph Construction

To detect communities, we construct a graph representation of the network and the different interactions between the different IP entities. More precisely, we rely on the multigraph structure since we need to consider multiple edges between the same pair of vertices, corresponding to different types of interaction.

The proposed framework is designed to model and analyze network dynamics by integrating temporal and structural data derived from network traffic. The framework captures the evolution of a graph G_t over time, resulting in a sequence of graphs $\{G_t\}$ that reflects the temporal evolution of the network communications.

We adopt a sliding window strategy over the network flows, ensuring that each graph $G_t(V_t, E_t)$ reflects the most recent traffic within a chosen interval. At every update:

1. **Collect flows** from the last window size W (*e.g.*, one hour).
2. **Rebuild or update** the multi-edge graph G_t, adding nodes and edges as described below.

3. **Run community detection** (label propagation) to assign node labels.

4. **Use a heterogeneous GNN** to identify anomalies.

Constructing the Time-Evolving Multi-Edge Graph

To capture changes in network communications, we construct a sequence of multi-edge graphs $\{G_t\}$, where each graph represents a time interval t.

Nodes: For each time t, V_t is the set of unique IP endpoints observed in the window $[t - W, t]$. Formally,

$$V_t = \{\text{SrcIP}, \text{DstIP} \mid \text{flows in } [t - W, t]\}. \tag{5.10}$$

Edges: E_t is a multiset of edges, allowing multiple distinct relationships (e.g., communication, context, knowledge) between the same pair of nodes (u, v). Each edge is labeled by its type k, reflecting the nature of the interaction:

$$\forall u, v \in V_t, \quad E_{uv,t} = \{\, e_k \mid e_k = (u, v, k) \wedge k \in \{\text{comm, context, knowledge}\}\}. \tag{5.11}$$

Hence, $E_t = \bigcup_{u,v} E_{uv,t}$.

At the beginning of each time interval, the multi-edge graph is constructed based on the current flow records in the buffer. The rolling window mechanism ensures that only flows within the window $[t-W, t]$ are used. Algorithm 7 outlines the steps to construct the time-evolving multi-edge graph.

Algorithm 7: Rolling Window Multi-Edge Graph Creation

Input: $\mathcal{B}_t$: buffer of flow records in $[t - W, t]$, W: window size, Δ_t: newly arrived flows in $[t - \Delta t, t]$
Output: $G_t(V_t, E_t)$: multi-edge graph for time t

Step 1: Update Buffer;
Remove records in $\mathcal{B}_{t-\Delta t}$ older than $t - W$;
$\mathcal{B}_t \leftarrow \left(\mathcal{B}_{t-\Delta t} \setminus \{r \mid r.\text{time} < t - W\}\right) \cup \Delta_t$;
Step 2: Build Multi-Edge Graph;
Initialize $V_t \leftarrow \emptyset$, $E_t \leftarrow \emptyset$;
foreach $r \in \mathcal{B}_t$ **do**
    $u \leftarrow r.\text{SrcIP}, \quad v \leftarrow r.\text{DstIP}$;
    $V_t \leftarrow V_t \cup \{u, v\}$;
    Insert Edge: $E_t \leftarrow E_t \cup \{(u, v, k)\}$, where k is based on flow characteristics:;
        if ***flow represents Network communication*** **then**
            set $k = \text{comm}$;
        else if ***flow provides contextual data*** **then**
            set $k = \text{context}$;
        else if ***flow involves knowledge-based interactions*** **then**
            set $k = \text{knowledge}$;
Define $G_t(V_t, E_t)$ as the multi-edge graph for time t;
return $G_t(V_t, E_t)$;

Valuation Function and Edge Types

We define a valuation function $W : E_t \to \mathbb{R}$ to compute weights for the edges in the multi-edge graph. The edges are partitioned into three types:

$$E_{\text{comm}}, \quad E_{\text{context}}, \quad E_{\text{knowledge}}.$$

The valuation function assigns a weight $W(u, v, k)$ based on the edge type k, as follows:

$$W(u,v,k) = \begin{cases} W^{(\text{comm})}(u,v,k), & (u,v,k) \in E_{\text{comm}}, \\ W^{(\text{context})}(u,v,k), & (u,v,k) \in E_{\text{context}}, \\ W^{(\text{know})}(u,v,k), & (u,v,k) \in E_{\text{knowledge}}. \end{cases} \tag{5.12}$$

Each weight $W^{(\cdot)}$ is computed as a weighted sum of normalized features, capturing specific attributes of the edge:

$$W^{(\text{X})}(u,v,k) = \sum_{i=1}^{m_{\text{X}}} \alpha_i^{\text{X}} \cdot \text{Norm}\big(x_i^{\text{X}}(u,v,k)\big), \tag{5.13}$$

where:

- m_{X}: Number of features for edges of the context X.
- $x_i^{\text{X}}(u, v, k)$: The i-th feature associated with the edge (u, v, k) of context `X`.
- $\text{Norm}(\cdot)$: A normalization function to scale features, such as min-max normalization.
- α_i^{X}: Coefficients representing the importance of each feature for context X.

This flexible weighting mechanism ensures that each edge type contributes meaningfully to the analysis, reflecting the nature of interactions captured by the graph.

5.4.1.3 Community Detection in the Graph of Interactions

After the graph of interactions has been constructed, we need to detect communities, *i.e.*, nodes that form a cluster because they exhibit similar communication patterns. The community detection algorithm has to provide near real-time calculation. We discard spinglass [217] and walktrap [218] algorithms since they cannot handle unconnected graphs. We compare the computation time of the following community detection algorithms:

Louvain [219] applies a greedy approach to construct communities based on a modularity metric. This metric compares the density of edges inside vs. outside the community.

Infomap [220] explores the graph with a random walk: a region where the random walker stays longer as statistically expected is considered a community.

LPA [221] (Algo. 8) assigns each node a label, updating them based on neighboring labels to form communities with strong internal connections.

Table 5.4: Comparison of Community Detection Algorithms on TON-IoT Dataset.

Algorithm	N. of Communities Detected	Processing time
Louvain	31	14.65 sec
Infomap	72	3028.03 sec
Label Propagation	48	1.7 sec

Table 5.4 illustrates the computation time of the different community detection algorithms with the TON-IoT dataset. The dataset includes diverse IoT traffic, including both benign and attack scenarios. Infomap is slow, taking over 50 minutes to identify communities. Louvain is faster but computationally expensive, taking around 15 seconds. Only Label Propagation Algorithm (LPA) can work in real-time, achieving close to 1s latency. The LPA technique is proposed as the best trade-off between community detection performance and execution speed. However, the approach is independent of the specific community detection algorithm used, so others may opt for higher accuracy in complex graphs at the cost of increased computational demands.

Community Detection via Label Propagation

For this task, we use LPA, which is well-suited for real-time applications and directed graphs.

After constructing G_t, we apply the LPA to detect communities within the graph. In our use case, the input to LPA includes the multi-edge graph G_t and initial labels for each node, typically set as the node's unique identifier.

The LPA updates node labels iteratively, adopting the most frequent label among neighbors. In directed multi-edge graphs, edge directionality is considered for neighbor relationships. The algorithm continues until labels stabilize, no further changes occur, and communities are identified.

Algorithm 8 provides the pseudocode for our adaptation of LPA to handle directed, multi-edge graphs efficiently.

The output of LPA is a set of community labels $\pi(v)$ for each node $v \in V_t$. Nodes sharing the same label are considered part of the same community. These labels are critical for the rest of the pipeline:

The HeteroGNN algorithm uses community labels as node features to enhance the graph's structural and contextual information, enabling better capture of node-edge relationships. The community structure provides insights into normal network behavior, highlighting anomalies as edges deviating from expected patterns. This step is crucial for improving the classification of edges as benign or anomalous, ensuring a comprehensive understanding of both structural and behavioral network patterns in the anomaly detection process.

5.4.1.4 Heterogeneous Graph Neural Network (HeteroGNN) for Anomaly Detection

The final stage of our anomaly detection pipeline employs a Heterogeneous Graph Neural Network (HeteroGNN) to classify network interactions (flows) as either be-

Algorithm 8: Label Propagation Algorithm for Community Detection

Input: Graph $G_t(V_t, E_t)$; maximum iterations $M_{\max}$
Output: A labeling function $\pi : V_t \to \mathcal{C}$ (community IDs)

Initialization:;
foreach $v \in V_t$ **do**
  $\pi(v) \leftarrow \text{uniqueLabel}(v)$;
Repeat up to $M_{\max}$ times:;
$changed \leftarrow false$;
foreach $v \in V_t$ ***(in random order)*** **do**
  Let $InNeighbors(v) = \{ u : (u \to v, k) \in E_t \}$; `// Collect inbound neighbors`
  if $InNeighbors(v)$ ***is not empty*** **then**
    $\ell \leftarrow$ most frequent label among $\{\pi(u) : u \in InNeighbors(v)\}$;
    if $\ell \neq \pi(v)$ **then**
      $\pi(v) \leftarrow \ell$;
      $changed \leftarrow true$;
if $changed = false$ **then**
  break;

nign or anomalous. IIoT environments exhibit complex interactions involving diverse devices and contextual dependencies. Standard GNNs, which assume homogeneous relationships, fall short in capturing the distinct types of interactions typical in IIoT networks. Therefore, a specialized heterogeneous graph approach is needed, enabling the differentiation and independent processing of multiple edge types (communication, context, and knowledge).

We specifically adopt a HeteroGNN due to its ability to independently process and combine different edge types. Traditional GNNs treat all edges equally, losing critical domain-specific distinctions. In contrast, HeteroGNNs use multiple convolutional layers tailored explicitly for each edge type, effectively capturing unique contributions of each relationship type [222]. By clearly distinguishing between different interactions, the model improves its anomaly detection accuracy, specifically in complex environments like IIoT networks where both structural and contextual anomalies occur.

Our heterogeneous graph G_t consists of nodes representing network hosts (devices) and multiple types of edges representing distinct interaction categories:

- **Communication Edges:** Represent direct data exchanges between devices, characterized by network traffic features (packet count, byte size, etc.).
- **Context Edges:** Encode contextual relationships based on temporal or operational similarities, capturing conditions such as operational timing, shared environmental conditions, or correlated behaviors.
- **Knowledge Edges:** Represent logical or policy-defined relationships between devices, derived from predefined operational rules or configurations (e.g., allowable protocol interactions, expected communication roles).

The HeteroGNN architecture applies edge-type-specific Graph Convolutional

Network (GCN) layers [223]. Mathematically, a single GCN convolutional operation for node i and edge type r is expressed as:

$$h_i^{(l+1,r)} = \sigma\left(\sum_{j\in N_r(i)} \frac{1}{\sqrt{|N_r(i)||N_r(j)|}} W^{(l,r)} h_j^{(l)}\right),$$

where:

- $h_i^{(l,r)}$ denotes the node embedding of node i for edge type r at layer l.
- $N_r(i)$ denotes neighbors of node i connected by edges of type r.
- $W^{(l,r)}$ is a learnable weight matrix specific to edge type r at layer l.
- $\sigma(\cdot)$ is an activation function (e.g., ReLU).

Separate `GCNConv` layers compute embeddings independently for communication, context, and knowledge edges. Each convolution captures the unique structural information from its corresponding edge type.

The input node features include:

- **Community Label:** Identifies the device community or cluster, capturing interaction patterns that are normal within communities but abnormal across boundaries.
- **Node Degrees (In-Degree and Out-Degree):** Representing the number of incoming and outgoing interactions, useful in detecting abnormal activities or attacks that cause unusual changes in device activity levels.

These features enhance the sensitivity of the model to structural and temporal anomalies, reflecting the dynamic nature of industrial IoT interactions.

As depicted in Figure 5.9, our model's workflow is structured as follows:

1. **Independent Convolutions:** Each edge type (communication, context, knowledge) is independently processed through dedicated GCN convolutional layers, generating specialized embeddings.
2. **Feature Refinement:** The embeddings are normalized using batch normalization layers and non-linear transformations (ReLU activation) to stabilize training and introduce non-linearity.
3. **Embedding Aggregation:** Refined embeddings from all edge types are aggregated through summation, forming comprehensive and unified node embeddings.
4. **Dimension Reduction and Classification:** A Multi-Layer Perceptron (MLP) reduces embedding dimensionality and extracts discriminative features. Edge classification is performed by concatenating source and destination node embeddings and passing them through an edge-specific classification head. The head outputs a probability indicating the likelihood of anomalous interaction.

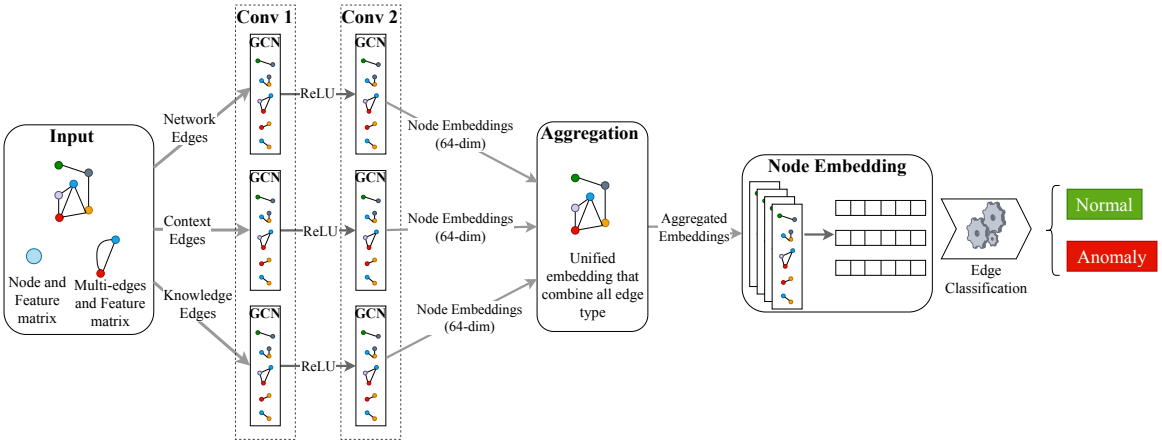


Figure 5.9: Our Proposed GNN Model Architecture.

The model is trained using a focal loss function [224], specifically chosen for its robustness in scenarios with class imbalance (common in anomaly detection tasks): $\text{FocalLoss}(p_t) = -\alpha_t(1-p_t)^\gamma \log(p_t)$, where p_t represents the predicted probability for the true class, α_t balances class weights, and γ focuses training on harder-to-classify samples.

Algorithm 9 summarizes our HeteroGNN model processing.

Algorithm 9: HeteroGNN for Edge Anomaly Detection

Input: Graph $G_t(V, E)$, Node features (community labels, degrees), Edge-type-specific adjacency matrices, Focal loss hyperparameters (α, γ)
Output: Edge anomaly predictions (probabilities)

foreach ***edge type*** $r \in \{$***communication, context, knowledge***$\}$ **do**
  Compute node embeddings with GCN convolution for type r using:

$$h_i^{(l+1,r)} = \sigma\left(\sum_{j \in N_r(i)} \frac{1}{\sqrt{|N_r(i)||N_r(j)|}} W^{(l,r)} h_j^{(l)}\right)$$

Apply batch normalization and ReLU activation to embeddings of each edge type;
Aggregate embeddings from all edge types via summation to obtain unified node embeddings;
Reduce dimensionality of embeddings using a Multi-Layer Perceptron (MLP);
foreach ***edge*** $(u, v) \in E$ **do**
  Concatenate embeddings of source node u and destination node v;
  Compute anomaly probability with edge-specific classification head;
Compute focal loss:

$$\text{FocalLoss}(p_t) = -\alpha_t(1-p_t)^\gamma \log(p_t)$$

Backpropagate loss and update model parameters.

By leveraging multiple graph convolutional layers tailored to different interaction types and incorporating contextual and structural features explicitly, our HeteroGNN approach significantly improves anomaly detection accuracy in com-

plex industrial IoT environments. Unlike homogeneous GNN models, our heterogeneous approach effectively captures subtle patterns indicative of anomalous behaviors across multiple dimensions of interactions.

5.4.2 Evaluation

We first evaluate the performance of our approach in isolation, and then compare our solution with the state-of-the-art techniques with two datasets. This evaluation emphasizes both accuracy and real-time responsiveness. The model was trained using a mobile computing platform Apple M2 Pro with 10 CPU cores and 32 GB RAM. To ascertain the robustness of our approach, the model underwent training over 50 epochs, allowing us to extensively evaluate its learning capability and performance stability over time.

5.4.3 Method

We generate time-based snapshots of the network to capture temporal dynamics and construct multi-edge graphs with distinct edge types (e.g., network communication, context, knowledge). Each node is labeled based on the community it belongs to, facilitating anomaly detection. Figure 5.10 illustrates our model, including all its components. This approach not only enhances the detection of network anomalies but also adapts to evolving traffic patterns, ensuring robust security in dynamic network environments.

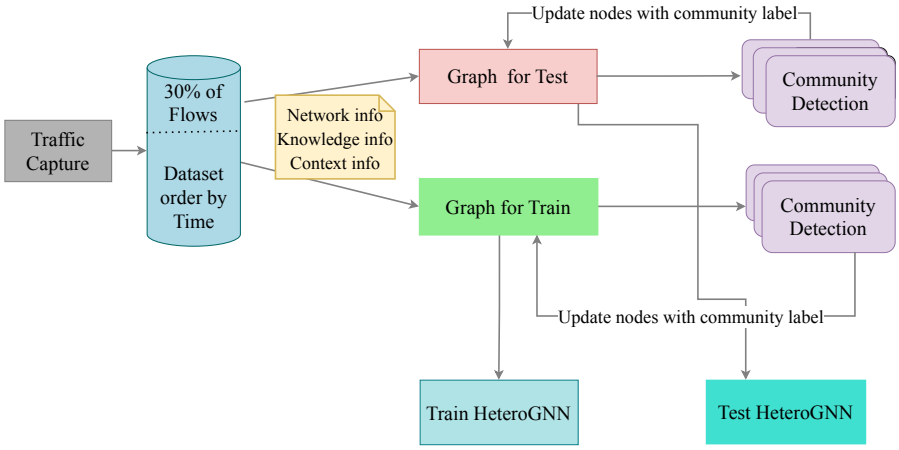


Figure 5.10: Integrated Framework for HeteroGNN Anomaly Detection.

5.4.3.1 Dataset Preparation

Our performance evaluation relies on the popular CIC-ToN-IoT [225] and CIC-IDS2017 [226] datasets. These datasets are widely used in the literature as they regroup both benign traffic and simulated IoT specific attacks. They mimic a realistic network environment by including telemetry data of IoT services, network traffic, and operating system logs.

To clarify the scale and difficulty of the classification task, we explicitly report the anomaly distribution in each dataset.

CIC-IDS2017: This dataset comprises 2,830,743 bidirectional flow records generated using CIC-FlowMeter. Among these, 557,646 records ($\approx$ 19.7%) are labeled as attacks. The remaining 2,273,097 records correspond to benign traffic. The attack flows span diverse categories, with the most prevalent being PortScan ($\approx 158,000$), DDoS ($\approx 162,000$), DoS-Hulk ($\approx 128,000$), and Brute-force (FTP + SSH, $\approx 13,000$), followed by Botnet ($\approx 1,900$), and a small number of Web and Infiltration attacks (fewer than 10,000 total).

CIC-ToN-IoT: We used the CIC-FlowMeter version of the dataset containing 1,846,373 bidirectional flow records. Of these, 461,934 records (25%) are labeled as attacks, while 1,384,439 are benign. The attack traffic includes DDoS (193,252), Scanning (105,699), Injection (72,534), Backdoor (19,126), DoS (19,243), Password cracking (15,277), MITM (1,348), and Ransomware (149).

Stratified sampling was used to preserve class proportions during training and testing splits (70/30), ensuring consistent evaluation conditions.

We chose these datasets to showcase our approach's ability to handle large-scale networks and process extensive data volumes in dynamic environments, effectively addressing diverse and complex cyber threats. The CICIDS2017 dataset comprises 10,000 nodes and 2.8 million edges. TON-IoT, on the other hand, boasts 5,000 nodes and approximately 19.4 million edges.

We acknowledge that the baseline methods used differ slightly between the CIC-IDS2017 and CIC-ToN-IoT evaluations. This decision was driven by compatibility constraints and implementation availability for certain models across specific dataset formats. For instance, DyEdgeGAT is designed for multivariate sensor time-series data and is not readily applicable to flow-based datasets such as CIC-IDS2017 without significant preprocessing or retraining. Despite this, we ensured that each dataset comparison included both graph-based and non-graph-based state-of-the-art models to support meaningful performance benchmarking.

5.4.3.2 Features Selection

In this study, we use the Gini index, a criterion in the Classification and Regression Trees (CART) algorithm, to identify relevant features for anomaly detection. It quantifies a node's impurity in a decision tree, helping distinguish normal from abnormal traffic. The features are ranked based on their importance to enhance the model's ability to recognize anomalies.

The Gini index for a set S containing classes C is formally defined as follows:

$$Gini(S) = 1 - \sum_{i=1}^{C} p_i^2 \tag{5.14}$$

where p_i is the proportion of samples in class i within the set S.

Relevant Features in the Dataset

The results presented in Figure 5.11 offer valuable insights into the key features for anomaly detection. Using the Gini Index, the analysis ranked feature importance from the CIC-ToN-IoT and CICIDS2017 datasets, identifying critical factors such as source and destination ports, forward packet length mean, and protocol-specific

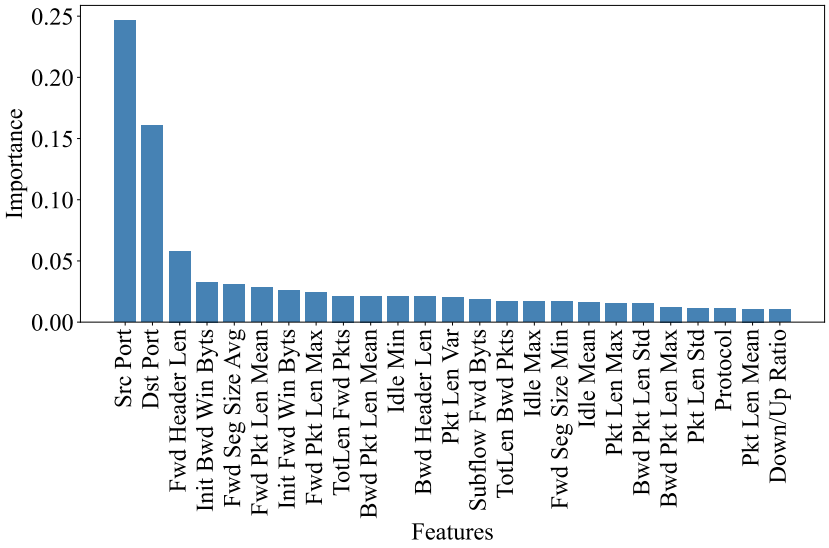


Figure 5.11: Top 25 Features Importance Based on Their Gini Index.

flags. These features highlight the significance of service-level attributes and communication patterns in distinguishing anomalous traffic. To enhance anomaly detection, the features were grouped into three categories for constructing multi-edge graphs:

1. Network Communication Features: Metrics like flow duration and packet counts represent communication intensity and patterns.

2. Contextual Features: Variables such as inter-arrival times and idle periods reveal behavioral and temporal anomalies.

3. Knowledge-Based Features: Attributes like packet size statistics and protocol flags encode logical relationships and expected norms.

This structured categorization facilitated the integration of feature importance into graph construction, improving the model's ability to detect diverse attack vectors across dynamic IoT networks.

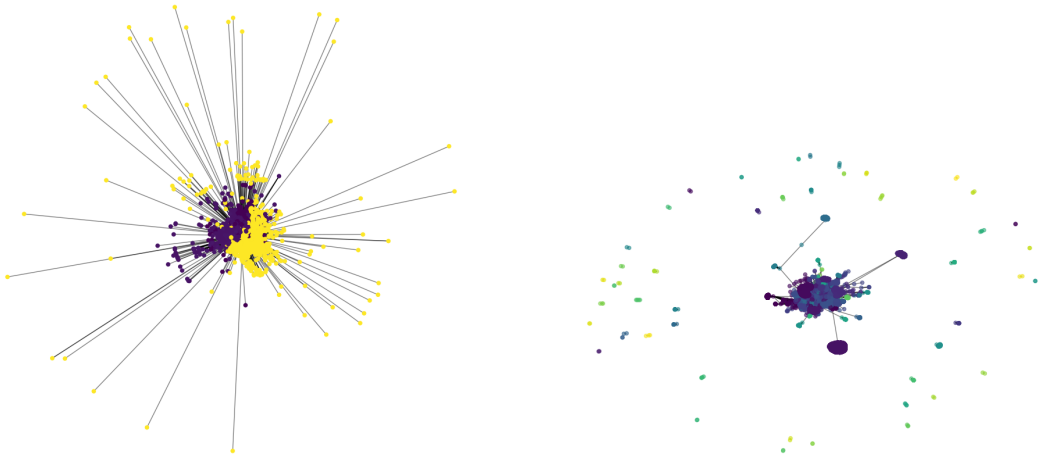
5.4.3.3 Evaluation and Metrics

Table 5.5 summarizes the architecture and training parameters used for the heterogeneous GNN model. These parameters are derived from the implementation, ensuring a balance between model complexity and computational efficiency. The configuration, including the use of Focal Loss and Adam optimizer, is specifically tailored to handle class imbalance and dynamic graph data, making the model robust for detecting anomalies in diverse IoT environments.

The model's performance is evaluated using Area Under the Curve (AUC), ROC curve as well as precision, recall and F1-Score.

Table 5.5: Summary of GNN Architecture and Training Parameters.

Parameter	**Value**
Hidden Channels	64
Activation	ReLU
Learning Rate	0.001
Weight Decay	10^{-3}
Optimizer	Adam
Loss Function	Focal Loss (with $\alpha = 0.25$, $\gamma = 2.0$)
Epochs	70



(a) CIC-IDS2017 (b) CIC-ToN-IoT

Figure 5.12: Network Graph of Communities.

5.4.3.4 Communities

We first illustrate in Figures 5.12a and 5.12b the communities identified in the two datasets used for the evaluation. In the CIC-IDS2017 dataset, the data is highly imbalanced, and the network lacks diversity in communication patterns (Figure 5.12a). The network exhibits a centralized structure, characterized by a small number of highly connected hubs and limited interconnectivity among other nodes. Specifically, only five communities in the network contain more than three nodes, indicating a sparse and hierarchical communication pattern.

In contrast, the CIC-ToN-IoT dataset exhibits a more balanced structure, with 48 communities with three or more members, indicating a broader, distributed network topology (Figure 5.12b). This network is characterized by more communities, higher interconnectivity, and less reliance on centralized hubs, resulting in a more decentralized and heterogeneous communication structure.

As it shows in Figures 5.16a and 5.16c the model's accuracy in distinguishing benign from attack edges is enhanced when community detection is included. This includes richer node features and embeddings, as shown in Figures 5.16a confusion matrix resulting in fewer false positives and false negatives. This leads to higher precision and recall. However, removing community information reduces the model's ability to correctly classify attacks due to the loss of helpful structural context.

Table 5.6: Comparison of Edge Configurations for Anomaly Detection.

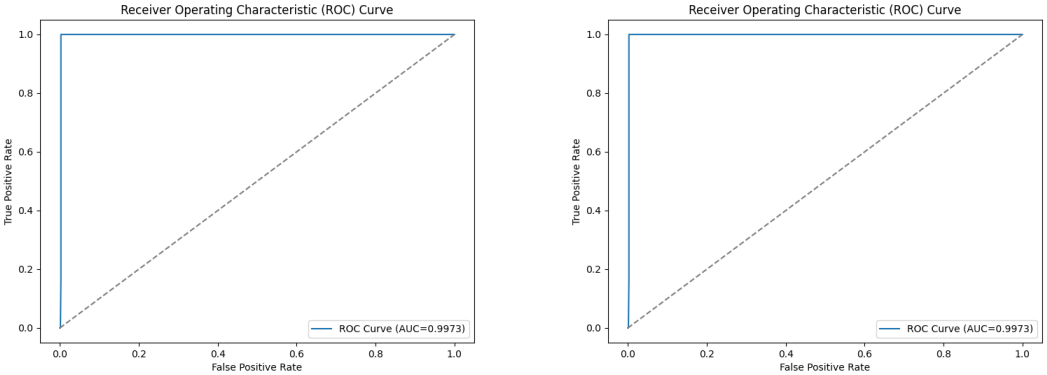
Edge Configuration	Precision	Recall	F1-Score	Accuracy	AUC
IDS2017 2 Edges	0.8996	1.0000	0.9472	0.9442	0.9973
IDS2017 3 Edges	0.9972	1.0000	0.9986	0.9986	0.9973
ToN 2 Edges	0.9778	1.0000	0.9888	0.9965	1.0000
ToN 3 Edges	0.9888	1.0000	0.9944	0.9982	1.0000
ToN No community	0.9615	0.8523	0.9036	0.9719	0.9922

5.4.3.5 Impact of the Number of Contexts

We first investigate the relevance of our context-aware design. Table 5.6 illustrates the efficiency of the method to detect anomalies when considering only 2 types of context (`Network Communication`, `knowledge`) vs. 3 types of context (`Network Communication`, `context`, `knowledge`).

In the CICIDS2017 dataset, precision increases from 89.96% in the 2-edge configuration to 99.72% in the 3-edge configuration, while the F1-score improves significantly from 94.72% to 99.86%. Similarly, for the CIC-ToN-IoT dataset, the precision improves from 97.78% to 98.88%, with the F1-score increasing from 98.88% to 99.44%. Importantly, recall remains consistently at 100% across all configurations, indicating the model's robustness in detecting true anomalies.

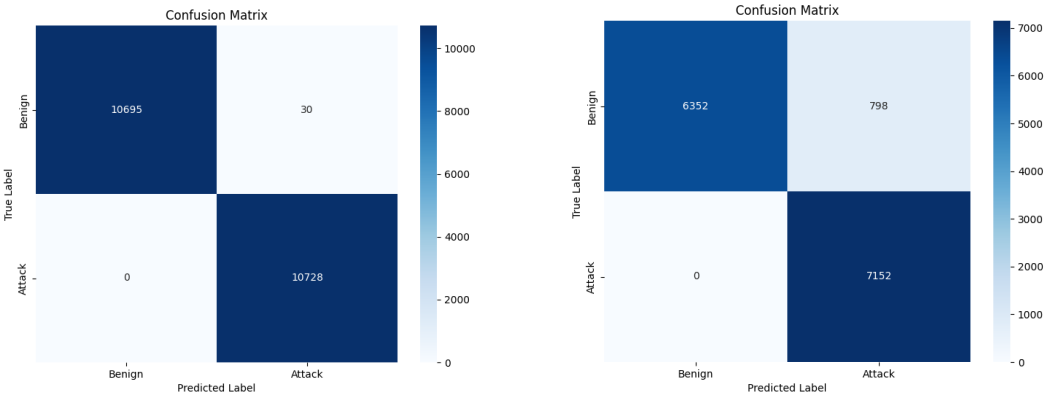
We can conclude that incorporating all contexts together enhances Precision, F1-score, and Accuracy. This demonstrates the agility of our architecture in accommodating multiple contexts and highlights the importance of context-aware decision-making in reducing false positives. Notably, this improvement in anomaly detection capability is achieved without compromising recall.



(a) 3 Edges (b) 2 Edges

Figure 5.13: Comparative ROC Curve for CIC-IDS2017 Dataset.

Figures 5.13a and 5.13b show the comparative ROC curves for the CIC-IDS2017 dataset. To further analyze these outcomes, Figures 5.14a and 5.14b present the corresponding confusion matrices. The 3-edge model exhibits fewer false positives and false negatives, illustrating its robust classification of normal versus attack traffic. While the 2-edge model remains competent, its slightly higher number of misclassifications underscores the benefit of leveraging a richer, more interconnected rep-

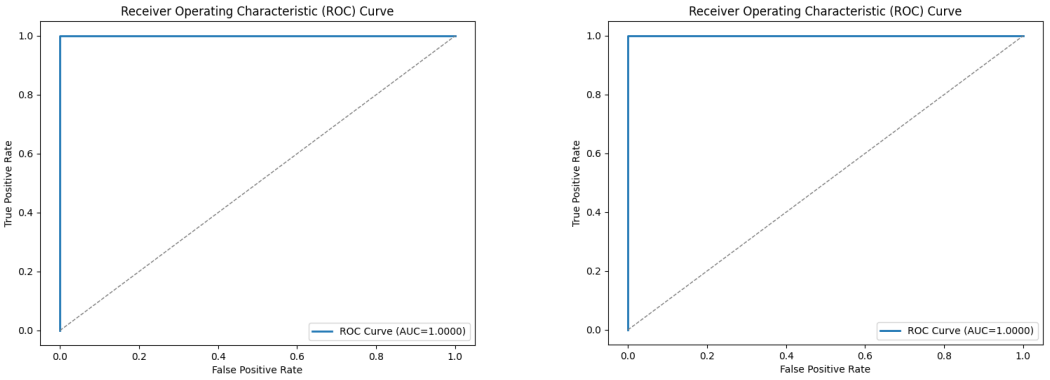


(a) 3 Edges (b) 2 Edges

Figure 5.14: Comparative Confusion Matrix for CIC-IDS2017 Dataset.

resentation in the 3-edge scenario. Hence, for this more heterogeneous dataset, the additional edge information clearly enhances the model's ability to detect anomalies accurately.

Both variants again perform well, with the 3-edge model achieving an F1 score of 0.9944 and the 2-edge model attaining an F1 score of 0.9888. Although the improvement of the 3-edge approach is not as dramatic as in the CIC-IDS2017 case, this narrower gap suggests that IoT traffic patterns in the CIC-ToN-IoT dataset are relatively straightforward to model. Still, capturing additional structural relationships through a third edge slightly strengthens the model's discrimination capability.

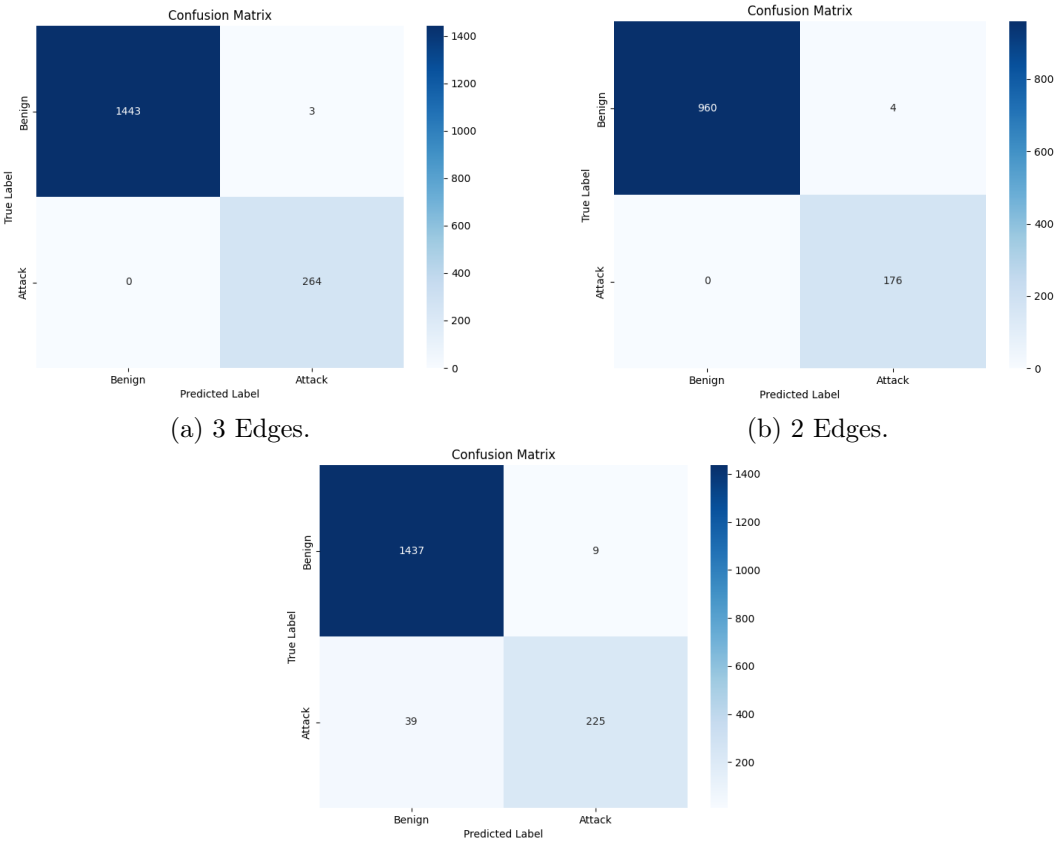


(a) 3 Edges (b) 2 Edges

Figure 5.15: Comparative ROC Curve for CIC-ToN-IoT Dataset.

The confusion matrices in Figures 5.16a and 5.16b confirm these observations, revealing that the 3-edge approach yields marginally fewer misclassifications than the 2-edge approach. This translates to more consistent detection of intrusions with minimal false alerts. Despite the reduced margin of difference, the results demonstrate that expanding the network graph representation from two edges to three edges can still confer measurable advantages in capturing subtle anomalies inherent to IoT systems.

Thus, our findings reinforce that the proposed 3-edge model outperforms the 2-



(a) 3 Edges.

(b) 2 Edges.

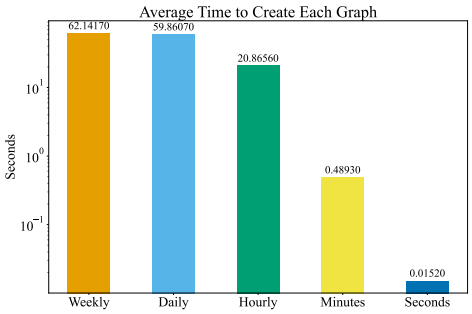
(c) 3 Edges Without Considering Community Detection.

Figure 5.16: Comparative Confusion Matrix for CIC-ToN-IoT Dataset.

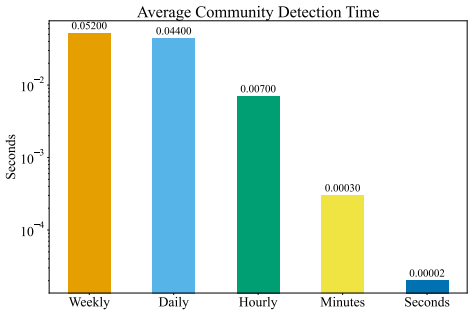
edge version across both benchmarks. The performance benefit is more pronounced in the CIC-IDS2017 dataset, which features a broader range of attack types and traffic patterns, thereby amplifying the value of additional structural information in the model. In contrast, the CIC-ToN-IoT dataset presents a narrower behavioral spectrum, but still benefits from more extensive graph connections. These results validate our hypothesis that incorporating a richer graph structure enables improved anomaly detection, particularly in complex network environments.

5.4.3.6 Impact of the Time Scale

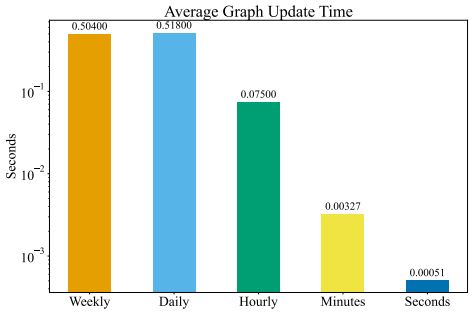
Based on the performance data provided (Figure 5.17), we can observe significant improvements in the system's efficiency when moving from larger time intervals (weekly) down to smaller intervals (seconds). This suggests that the system is highly optimized for real-time processing, particularly in scenarios requiring frequent updates. The performance data reveals a clear trend of decreasing execution times across various tasks as the time intervals shorten from weekly to secondly. Graph creation time drops from 62.1417 seconds weekly to just 0.0152 seconds at the secondly level. Similarly, label propagation time decreases from 0.052 seconds to 0.00002 seconds, and graph update time reduces from 0.504 seconds to 0.00051



(a) Average Time to Create Each Graph.



(b) Average Community Detection Time.



(c) Average Graph Update Time.

Figure 5.17: Performance Trends Across Different Time Scales.

seconds.

A crucial measure is the test time of 0.0075 seconds, which represents the time taken to run the GNN model and determine whether the given traffic is anomalous. This demonstrates the system's real-time capability in anomaly detection, processing all operations swiftly and efficiently, making it well-suited for environments requiring rapid, high-frequency updates with minimal delay.

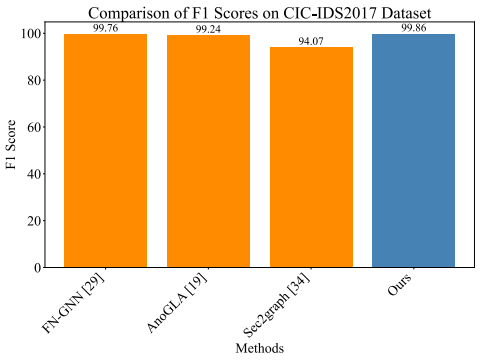
5.4.3.7 Comparison With State-Of-The-Art Techniques

We compare our proposed AE-LDA model against existing state-of-the-art anomaly detection methods evaluated on the CIC-ToN-IoT and CIC-IDS2017 datasets. To ensure fairness and reproducibility, we did not re-implement competing methods, but instead directly report their published performance metrics from existing peer-reviewed studies.

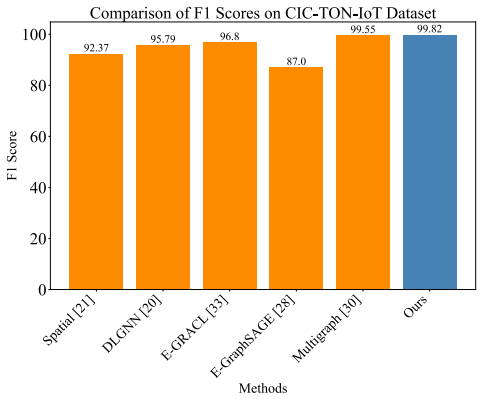
Performance comparisons focus exclusively on AUROC scores, as they provide a standardized and widely accepted measure for anomaly detection capability. While hardware specifications, such as CPU and GPU details, are reported here for completeness and transparency regarding our own implementation environment (Intel Xeon Gold 6330 @ 2.00 GHz, RTX 3090 GPU, and 24 GB of memory), these characteristics do not directly affect our comparative analysis, since our primary objective is evaluating predictive accuracy rather than computational efficiency.

CIC-ToN-IoT Dataset

Figure 5.18b contrasts the F1 scores of multiple anomaly detection approaches applied to the CIC-ToN-IoT dataset. We incorporate results for Spatial [227], DLGNN [228], E-GRACLN [112], E-GraphSAGE [111], and Multigraph [113]. Notably, E-GraphSAGE attains a commendable F1 score of 96.8%, highlighting the value of graph-based message passing in capturing relationships among IoT nodes. Meanwhile, DLGNN (95.79%) and Spatial (92.37%) demonstrate robust but slightly lower performance, suggesting that while these architectures handle IoT-related features effectively, they may miss certain deeper relational or contextual cues. Multigraph achieves a particularly high F1 score of 99.55%, indicating that additional layers of graph abstraction can significantly boost detection capabilities in IoT settings.



(a) CIC-IDS2017



(b) CIC-ToN-IoT

Figure 5.18: Comparison of Different Methods.

CIC-IDS2017 Dataset

Figure 5.18a illustrates the F1 performance on the CIC-IDS2017 dataset for our approach in comparison with FN-GNN [229], AnoGLA [230], and Sec2graph [110]. Both FN-GNN and AnoGLA surpass 99% F1, reflecting their sophisticated graph-based learning and attention mechanisms tailored for cybersecurity data. By contrast, Sec2graph achieves 94.07%, which is competitive but indicates potential limitations in capturing a broader range of threat behaviors within enterprise traffic. Our approach reaches an F1 score of 99.86%, improving slightly over FN-GNN and AnoGLA, and markedly exceeding Sec2graph.

While CIC-ToN-IoT traffic often displays more homogeneous, "community-like" structures where device behaviors and protocols follow narrower patterns, the CIC-IDS2017 dataset encompasses diverse enterprise traffic spanning multiple services, protocols, and user behaviors. Such variability can make classification more challenging, as malicious flows might resemble legitimate but less common activities. Nonetheless, our context-aware and community-enhanced graph construction proves equally effective here. By identifying cohesive sub-networks and analyzing node-level anomalies within these communities, our method achieves robust detection despite the dataset's more heterogeneous nature.

In Figure 5.18a, the IoT-based dataset (CIC-ToN-IoT) shows more consistent, community-like structures (see Figure 5.12) than the diverse enterprise-style traffic in Figure 5.18b (CIC-IDS2017). This tighter distribution of "normal" IoT traffic makes anomalies stand out more distinctly, leading to better classification performance. By contrast, enterprise traffic encompasses various protocols and behaviors, causing more overlap between benign and malicious samples. Additionally, the flow-based time granularity in both figures, typically on the order of 1–5 seconds, ensures that each node or point represents an aggregated snapshot of network activity, further emphasizing the clear structural patterns in the IoT dataset.

Finally, the high F1 scores observed underscore how effectively our approach balances precision and recall. Particularly in security contexts, an accurate detection system must not only identify threats but do so with minimal false alarms. Excessive false positives can overwhelm analysts, while false negatives allow intrusions to remain undetected. That our model consistently reports near-ideal F1 scores (above 99%) in two very different environments speaks to its versatility and potential readiness for production environments. These outcomes lay a foundation for future work aimed at extending the framework to more granular real-time intrusion detection, cross-dataset transfer learning, or distributed implementations that further reduce the computational overhead on any single node.

5.5 Conclusion

This chapter introduced two contributions aimed at addressing critical aspects of anomaly detection and assessment within IIoT networks, emphasizing the importance of real-time detection to ensure robust security in industrial environments. The first contribution, "Proposed Network Anomaly Detection," combined deep-learning autoencoders with linear discriminant analysis, significantly enhancing anomaly detection by effectively identifying previously unknown network threats, including zero-day attacks.

The second contribution, "Context-Aware Behavioral Anomaly Detection," advanced the anomaly detection process by integrating contextual insights and community-based graph detection techniques. Unlike traditional methods, this approach identifies anomalies by examining changes in device interactions, effectively capturing complex behavioral patterns.

However, Despite demonstrating robust performance, the AE-LDA anomaly detection approach is limited by its focus on network features alone – thereby neglecting critical contextual cues needed for a comprehensive analysis. Additionally, the context-aware behavioral approach relies heavily on the quality and availability of accurate contextual data, which may pose challenges in dynamic or unpredictable industrial settings.

This chapter introduced advanced anomaly detection methods tailored explicitly to industrial IoT environments, demonstrating their effectiveness in reliably identifying security threats and anomalies in complex network structures. By effectively distinguishing normal operational variations from genuine attacks, these anomaly detection approaches provide a crucial foundational layer of security.

Building upon this robust anomaly detection foundation, the next phases of this thesis extend the scope of investigation to integrated and dynamic security solu-

tions. Specifically, we explore how anomaly detection outcomes can be seamlessly integrated with blockchain technologies for transparent, immutable security monitoring, and how zero trust principles can further enhance security by continuously validating interactions within industrial networks. These advanced approaches aim to provide comprehensive, resilient, and adaptable cybersecurity solutions suited for evolving industrial environments.

Chapter 6

Blockchain Approach for Securing Distributed Industry Environments

Contents

Publications:

- Fatemeh Stodt, Mohammed BM Kamel, Christoph Reich, Fabrice Theoleyre, and Peter Ligeti. "Blockchain-Based Privacy-Preserving Shop Floor Auditing Architecture". In: ***IEEE Access*** 12 (2024), pp. 26747–26758
- Fatemeh Stodt, Mohammed Alshawki, Christoph Reich, Peter Ligeti, and Fabrice Theoleyre. "Securing the Future: Lightweight Blockchain Solutions for IIoT and IoT Networks". In: ***Security and Privacy*** 8.4 (2025), e70070
- Fatemeh Stodt, Philipp Ruf, and Christoph Reich. "Blockchain-Enabled Digital Product Passports for Enhancing Security and Lifecycle Management in Healthcare Devices". In: ***2024 8th Cyber Security in Networking Conference (CSNet)***. IEEE. 2024, pp. 44–51

6.1 Introduction

Securing distributed industrial systems presents unique challenges due to their large-scale, heterogeneous, and resource-constrained nature. While traditional security mechanisms (such as centralized identity management and perimeter-based access control) have played a crucial role in enterprise IT, they often fall short in IIoT environments where decentralization, trust minimization, and auditability are essential. In particular, the need for real-time response, secure data provenance, and tamper-proof interactions among untrusted devices highlights the inadequacy of conventional approaches.

Blockchain technology has emerged as a promising enabler for secure, transparent, and decentralized IIoT environments. Its inherent properties (immutability, distributed consensus, and traceability) align well with the security and operational demands of critical industrial systems. However, integrating blockchain into IIoT ecosystems introduces new constraints, especially regarding scalability, energy efficiency, and support for low-power devices. Addressing these constraints requires tailored architectures and lightweight consensus mechanisms.

This chapter introduces two complementary blockchain-based contributions that aim to meet these challenges:

- **Shopfloor Blockchain Approach** (Section 6.3): This architecture provides privacy-preserving authentication and auditable data flows for industrial shopfloor environments, using an attribute verification protocol and a multi-tiered blockchain structure.
- **Lightweight Blockchain Approach** (Section 6.4): This contribution proposes a scalable and energy-efficient blockchain solution for IIoT networks. It overcomes the limitations of traditional blockchain implementations (such as high computation, communication, and storage overhead) by using a Byzantine Fault Tolerance (BFT)-Directed Acyclic Graph (DAG) and role-based node architecture.

Together, these contributions address key research challenges related to scalability, privacy, and real-time security assurance in IIoT networks. By tailoring blockchain mechanisms to the constraints and needs of industrial settings, this work advances the feasibility of secure, distributed, and privacy-preserving IIoT infrastructures.

Addresses Research Questions:

- **How can blockchain address IIoT challenges of scalability, privacy, and tamper-proofing?**

6.2 Problem Statement

IIoT networks continue to face significant security challenges, largely due to their inherent complexity, distributed structure, and vulnerability to evolving threats [197, 198]. Traditional security frameworks frequently lack essential properties such as

decentralization, transparency, and immutability, limiting their effectiveness in safeguarding industrial processes against sophisticated attacks [231].

Blockchain technology has emerged as a potential solution to address these shortcomings by offering decentralized, cryptographic security with transparent and immutable records [232]. Nonetheless, integrating blockchain into IIoT introduces substantial challenges that existing blockchain approaches have struggled to fully address:

- **Scalability and Real-Time Performance:** Existing blockchain implementations often suffer from latency issues and computational overhead, limiting real-time applicability in high-throughput industrial settings [133, 140].
- **Privacy and Security:** Many blockchain-based IIoT solutions lack privacy-preserving mechanisms, potentially exposing sensitive data to unauthorized access or privacy breaches [141].
- **Resource Constraints:** Typical blockchain solutions have substantial computational and storage requirements, posing deployment challenges in resource-constrained industrial environments [141].
- **Resilience Against Adversarial Attacks:** Existing implementations rarely consider real-world industrial network conditions characterized by adversarial threats, such as DoS attacks, and unpredictable network dynamics [140].

Therefore, a blockchain framework tailored explicitly to IIoT contexts must efficiently manage computational load distribution, provide privacy-preserving mechanisms, and ensure scalability under real-world industrial conditions. This chapter proposes such an approach, introducing a hierarchical blockchain architecture combined with a Practical Byzantine Fault Tolerance (PBFT)-based consensus mechanism and attribute-based verification for privacy-preserving authentication.

Lightweight Blockchain for IIoT Networks and Existing Limitations: IIoT environments are characterized by resource-constrained devices with strict computational, storage, energy, and communication limitations. Consequently, standard blockchain technologies, with their significant computational demands, large storage requirements, and energy-intensive operations, are unsuitable for direct deployment in these environments [147, 150]. Thus, implementing lightweight blockchain solutions that minimize computational load, network overhead, and storage requirements while ensuring high throughput and energy efficiency becomes crucial for industrial scenarios [151].

However, current lightweight blockchain solutions still face several significant challenges. Techniques such as block header collapsing, although reducing computational overhead, do not adequately address the network traffic and ledger-growth issues intrinsic to industrial-scale IoT deployments [150, 157]. Furthermore, consensus mechanisms like Proof of Work (PoW) are inherently unsuitable for energy-sensitive IIoT devices, and existing alternatives such as Proof of Authority (PoA) often compromise decentralization and introduce potential security risks [147].

To address these limitations, a viable lightweight blockchain architecture for IIoT must fulfill several essential requirements. First, it should ensure ***reduced computational load***, enabling lightweight processing suitable for devices with limited

computing power. Second, it must provide ***minimal network overhead*** through efficient data transmission protocols that reduce bandwidth consumption. Third, ***storage efficiency*** is necessary, with strategies to control ledger growth and optimize data storage on resource-constrained devices. Fourth, the system must support ***high throughput*** to ensure scalability and responsiveness under industrial workloads. Fifth, it should promote ***energy efficiency***, utilizing low-energy consensus mechanisms that are compatible with energy-limited IIoT nodes. Finally, maintaining ***decentralization and robustness*** is critical to resist node compromise and adversarial threats without relying on centralized control.

This chapter introduces a novel lightweight blockchain framework specifically designed to overcome these critical challenges, incorporating BFT consensus with dynamically verifiable nodes, hierarchical node roles for task distribution, and optimized ledger management strategies tailored explicitly for industrial IoT conditions.

6.3 Shopfloor Blockchain Approach

6.3.1 Preliminaries

6.3.1.1 Attribute Verification Protocol

Our blockchain architecture leverages an *attribute verification protocol* [233] as a foundational mechanism for privacy-preserving authentication and verification. This distributed protocol verifies participants based on their attributes without unnecessarily revealing sensitive information. Originally introduced for applications such as data validation [234] and network security [235], this protocol consists of three specific roles with clearly defined responsibilities:

- **Issuer:** An authoritative entity responsible for managing and distributing user attributes. The issuer provides public keys to verifiers and securely delivers attribute credentials to provers.
- **Verifier:** An entity that challenges provers to validate their claimed attributes in a zero-knowledge setting without compromising privacy.
- **Prover:** A participant who must demonstrate ownership of specific attributes. After successfully verifying ownership with the issuer, the prover receives a secret key enabling it to respond securely to verification challenges.

The attribute verification protocol supports two primary modes of verification:

- 1-***out-of-***n ***Verification Mode:*** The verifier defines a set of acceptable attributes. The prover successfully verifies its identity by demonstrating ownership of at least one attribute from this set, preserving privacy by not disclosing specific attribute details.
- n-***out-of-***n ***Verification Mode:*** All specified attributes must be verified. This mode provides comprehensive verification when strict attribute validation is required.

The 1-*out-of-n* mode is particularly relevant for IIoT contexts, as it maintains the privacy of industrial participants by minimizing attribute disclosure to only the essential verification criterion.

6.3.1.2 Architecture Preliminaries

The architecture employs a multi-tiered network structure, each tier with distinct roles and responsibilities:

- **Local Nodess (LNs):** Sensors and actuators with limited computational capacity are used for data collection and actuation tasks, relying on Middle Nodes (MNs) for secure communication and data storage. These devices operate under low-power conditions and have limited storage capacity.
- **Middle Nodes (MNs):** MNs optimize network data processing and storage by acting as bridges within subnetwork entities. Equipped with HSM, they generate cryptographic keys, maintaining data security and privacy. Strategically placed, MNs balance load and optimize network traffic.
- **Full Nodes (FNs):** FNs are the backbone of the blockchain, responsible for maintaining consensus and appending validated blocks. Strategically placed, they execute the Practical Byzantine Fault Tolerance algorithm, ensuring blockchain integrity and robustness. Their computational capability and reliability are chosen.

6.3.2 Architecture Description

The proposed system uses a harmonious collaboration between FNs and MNs, with MNs acting as local data keeper and facilitating computation offloading. A PBFT consensus mechanism by FNs ensures transaction reliability and security. The selection of FNs and MNs is based on computational capability, network connectivity, and trust level based on historical performance. As shows in Fig. 6.1, the method introduces "middle nodes" to interconnect isolated subnetworks, crucial for computational tasks and network efficiency.

The operation of the architecture can be understood as a series of coordinated steps. First, in the ***data acquisition*** phase, LNs capture real-time data and forward it to their corresponding MNs. Next, during ***secure storage***, the MNs store incoming data either directly or as cryptographic hashes to ensure data secrecy and integrity, examples include machine operational hours or threshold-based indicators such as temperature readings.

Following storage, the ***transaction lifecycle*** begins as middle nodes generate transactions from the collected data, sign them cryptographically, and propagate them to other MNs within the subnetwork to support redundancy and data availability. In the ***candidate block formation*** phase, these MNs compile verified transactions into candidate blocks, preparing them for broader network verification.

The next phase is ***network-wide verification***, where candidate blocks are broadcast to all LNs and validated using a PBFT consensus mechanism to ensure integrity and prevent data manipulation. Finally, during the ***blockchain append***

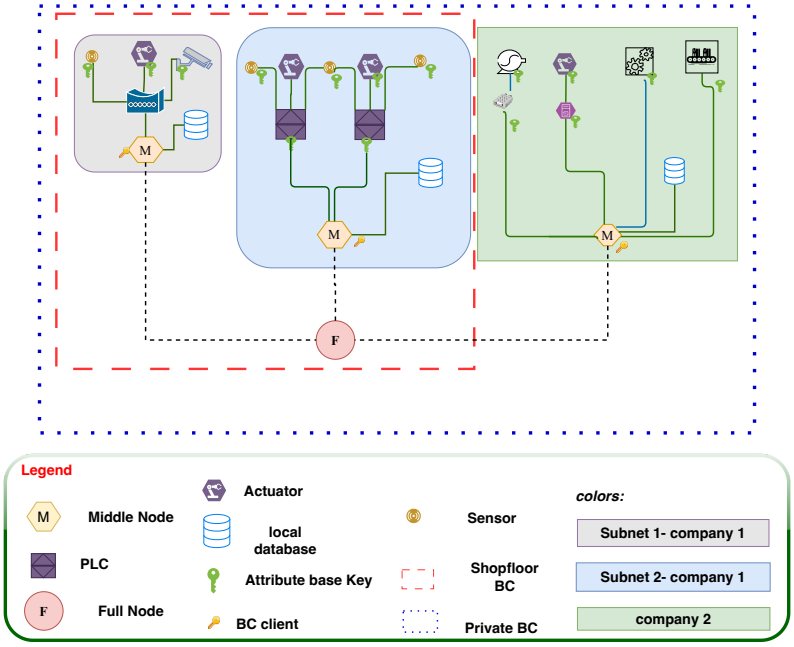


Figure 6.1: Shop Floor Blockchain Architecture.

phase, LNs combine the verified candidate blocks and add them to the blockchain, making the data permanent and tamper-resistant.

The complex design of LNs, which may include simple sensors, ensures secure and reliable data transfer. Meanwhile, MNs serve as buffering points, particularly valuable during network instability, safeguarding data and preparing it for seamless blockchain integration.

6.3.2.1 Architecture Workflow

The proposed architecture prioritizes participant privacy through an attribute verification protocol that operates in four phases: setup, registration, generation, and validation. MNs handle data encryption using HSMs keys for secure transaction initiation. LNs provide hashed data to MNs, adding another layer of privacy. Before participating in the subnetwork, participants must pass the preregistration step.

Setup Phase

The setup phase is a very critical pre-registration step in network configuration, during which an IIoT node is identified and integrated into the network. In this phase, and during registering a new node, the responsible LN assigns a unique identifier to guarantee its unique location inside the network. This is a dual-faced unique ID (ID_n) derived from the node's serial number (Sn) and MAC address (MAC) as in Equation 6.1. Whereas the MAC address is unique and identifiable in the digital world, the serial number is like an unchangeable imprint from genesis.

$$ID_n = H(Sn \parallel MAC) \tag{6.1}$$

Here, H represents a cryptographic hash function and $\|$ denotes concatenation. MAC, inherently unique and recognizable in the digital domain, serves as a reliable hardware-based identifier. In contrast, Sn acts like an immutable tag assigned at the time of manufacture. This dual-component approach to node identification enhances security, as it couples a physically unalterable attribute Sn with a digitally unique identifier MAC.

Algorithm 10: Setup Phase

Input: *node*: The node to be initialized
Output: *node*: The node with its unique ID
1 **Function** ***SetupPhase(****node****)*****:**
2 $ID_n \leftarrow$ `HashFunction`$(node.Sn \| node.MAC)$;
3 $node.unique\_ID \leftarrow ID_n$;
4 **return** *node*;

The setup phase (see Algorithm 10) is rigorously conducted only once for each node to maintain the integrity of these identifiers. During the identifier generation process, MAC and Sn data are retrieved from the newly joined node, playing a crucial role in the network's security architecture. These unique and immutable identifiers form a strong identification system for safe and effective network operations.

Registration Phase

During the setup phase, nodes are assigned a unique identity and are further characterized by configurable attributes that dictate their operational behavior and network interaction. Four primary attributes are discussed:

- **Logical Network Sector (SEC):** The attribute defines a node's operational scope within the network topology, determining its functional area and interaction with other network segments, aiding in effective segmentation and management.
- **Installer Signature (INS):** The INS is responsible for commissioning and configuring the node, ensuring traceability and accountability by reliably logging any modifications or installations.
- **Power Consumption (POC):** The LNs energy usage is crucial for network sustainability and efficiency, and monitoring POC helps optimize energy consumption and manage the environmental footprint of network operations.
- **Transmission Pattern (TRA):** The TRA is a crucial tool for managing network traffic, load balancing, and optimizing bandwidth usage by predicting and shaping the network's data flow.

The attributes of a system are dynamic and can be updated or expanded by MNs, making them crucial in a network environment that may evolve or require adjustments due to new operational demands or technological advancements. The *n-out-of-n* verification mode, as described in Section 6.3.1.1, is used for validation and tokenization of nodes based on their attributes.

1. **Attribute Verification Function:** we can represent the verification process of a node's attributes by the MNs as follows:

$$V_{attr}(LN) = \begin{cases} 1 & \text{if } SEC, INS, POC \text{ is valid} \\ 0 & \text{otherwise} \end{cases} \tag{6.2}$$

   Where $V_{attr}(LN)$ represents the verification function for the LN attributes. The function returns 1 if all the attributes (SEC, INS, POC, TRA) are successfully verified by the MNs, and 0 otherwise.

2. **Token Issuance Function:** After the verification, tokens are issued to certify the readiness of the LNs for network participation.

$$T_{issue}(LN) = \begin{cases} \text{Token} & \text{if } V_{attr}(LN) = 1 \\ \text{No Token} & \text{if } V_{attr}(LN) = 0 \end{cases} \tag{6.3}$$

   In equation 6.3, $T_{issue}(LN)$ represents the token issuance function. A token is issued if the verification function $V_{attr}(LN)$ returns 1, indicating successful verification of the node's attributes.

3. **Dynamic Attribute Update Function:** we can model the ability of MNs to update the attributes of LNs over time:

$$U_{attr}(LN, new\_attr) = \begin{cases} \text{Updated Attr} & \text{if update} \\ \text{Unchanged Attr} & \text{otherwise} \end{cases} \tag{6.4}$$

   Here, $U_{attr}(LN, new\_attr)$ denotes the attribute update function, where LN, new_attr represents the new attributes to be assigned to the LN. This function reflects the dynamic adaptability of the network's attributes.

The verification and tokenization of network attributes allows for granular control, enabling administrators to implement policies based on node characteristics, enhance security protocols, and optimize network performance. The registration phase (see Algorithm 11) is a crucial component in establishing a robust, efficient, and secure network infrastructure.

Algorithm 11: Registration Phase

Input: *node*: The node to be registered
Output: *node*: The node with updated attributes

```
Function RegistrationPhase(node):
    node.attributes ←
      {"SEC": None, "INS": None, "POC": None, "TRA": None};
    UpdateNodeAttributes(node);
    return node;
Function UpdateNodeAttributes(node):
    // Update attributes based on network policies
```

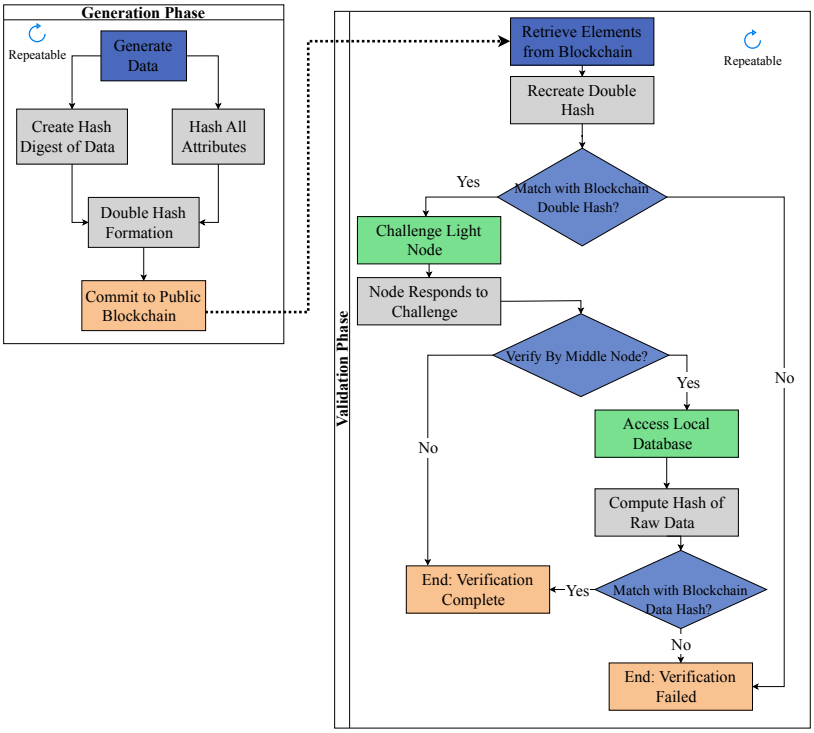


Figure 6.2: Generation and Validation Phases.

Generation Phase

The Generation Phase is a crucial stage in blockchain data management, involving the creation and preparation of data for entry into the blockchain, ensuring data integrity and security through sequential steps:

1. **Data Generation:** Initially, raw data is generated by LNs. This data could represent transactions, sensor outputs, user actions, or any relevant information that needs to be recorded on the blockchain.

$$D = \text{GenerateData}(\text{Raw Inputs}) \tag{6.5}$$

   Where D denotes the generated data from raw inputs.

2. **Data Digest Creation:** Each piece of data D is then processed to create a cryptographic hash digest. This digest serves as a unique fingerprint of the data.

$$H(D) = \text{Hash}(D) \tag{6.6}$$

   Here, $H(D)$ is the hash digest of data D.

3. **Attribute Digest Generation**: Concurrently, all relevant attributes of the data are hashed to ensure that every characteristic is accounted for and secured.

$$H(A) = \text{Hash}(A_1, A_2, \ldots, A_n) \tag{6.7}$$

   With $H(A)$ representing the combined hash of attributes $A_1, A_2, \ldots, A_n$.

4. **Double Hash Formation:** To further enhance security, a double hash is formed, which is the hash of the hash digest and the attribute digest.

$$H_2(D, A) = \text{Hash}(H(D) \parallel H(A)) \tag{6.8}$$

$H_2(D, A)$ symbolizes the double hash, where $||$ denotes concatenation.

5. **Commit to Public Blockchain:** The final step is committing the double hash to the public blockchain. This action immutably records the data and its attributes, ensuring traceability and verifiability.

$$B = \text{CommitToBlockchain}(H_2(D, A)) \tag{6.9}$$

Where B is the blockchain record containing the double hash $H_2(D, A)$.

The Generation Phase (see Algorithm 12) involves repeatable and scalable steps, ensuring the system can handle increased data generation without compromising security or integrity. The process emphasizes repeatability, allowing for efficient and secure data generation and recording in a consistent manner.

Algorithm 12: Generation Phase

Input: LN: Local Node containing raw inputs and attributes
Output: B: Blockchain commitment of the generated data

```
Function GenerationPhase(LN):
    D ← GenerateData(LN.raw_inputs);
    hash_digest ← HashFunction(D);
    attribute_hash ← HashFunction(LN.attributes);
    double_hash ← HashFunction(hash_digest, attribute_hash);
    B ← CommitToBlockchain(double_hash);
    return B;
```

The Generation Phase of blockchain technology ensures data protection against tampering and unauthorized modifications, upholding the principles of decentralization and trust.

Validation Phase

The Validation Phase in a blockchain-based network is a critical multi-tiered process that ensures data integrity, privacy, and compliance with network protocols. This phase involves several key steps:

1. **Initial Verification by MNs:** MNs verify LNs properties against pre-registered attributes to ensure each LN complies with network standards and policies. The process can be represented as follows:

$$V_{MN}(LN) = \begin{cases} 1 & \text{match pre-registered values} \\ 0 & \text{otherwise} \end{cases} \tag{6.10}$$

Where $V_{MN}(LN)$ is the validation function performed by the MNs on the LNs.

2. **Consensus Process by FNs:** To ensure the integrity and reliability of the blockchain's contents, FNs examine encrypted data blocks using the PBFT consensus method. The consensus can be represented as:

$$C_{PBFT}(Block) = \frac{\sum_{i=1}^{n} V_{FN_i}(Block)}{n} \tag{6.11}$$

Where $C_{PBFT}(Block)$ is the consensus function, $V_{FN_i}(Block)$ is the validation function performed by each FN on the block, and n is the total number of FNs participating in the consensus process.

3. **Random Audits for Data Integrity and Privacy:** Independent auditors conduct random checks post-verification to ensure data integrity and validate privacy-preserving measures (see Fig. 6.2), adding an additional layer of security and compliance verification.

The system combines privacy protocols with an efficient blockchain architecture, providing a robust privacy framework without compromising network speed and efficiency, a crucial aspect for adoption in sensitive environments like IIoT networks where data privacy and rapid processing are essential. The Validation Phase algorithm (see Algorithm 13) is defined as:

Algorithm 13: Validation Phase

Input: LN: Local Node, MNs: Monitoring Nodes, FNs: Functional Nodes, $auditors$: Auditing Nodes
Output: Validation result (True or False)

```
Function ValidationPhase(LN, MNs, FNs, auditors):
    if not Verify(MNs, LN) then
        return False;
    block ← LN.generate_block();
    if not PBFTConsensus(FNs, block) then
        return False;
    if not RandomAudit(auditors, LN) then
        return False;
    return True;
Function PBFTConsensus(FNs, block):
    // Implement PBFT consensus mechanism
    return True or False;
```

6.3.3 Analysis

The proposed architecture's effectiveness, security, and performance were assessed using an experimental research method, with an extensive analysis provided in this section.

6.3.3.1 Implementation and Evaluation Overview

The study used the Charm framework [236], for attribute verification in a test environment with a middle node and seven local nodes, each with a 1.8 GHz processing

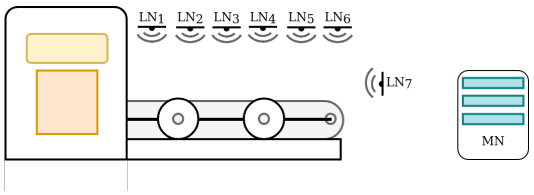


Figure 6.3: Implementation of the Proposed Framework.

unit, as illustrated in Figure 6.3. The evaluation focused on registration, generation, and validation phases, aiming to measure efficiency and system resilience against security threats. During the registration phase, a single MN is responsible for issuing tokens associated with four predefined attributes in the system: SEC, INS, POC, and TRA. Consequently, each LN receives four tokens from the MN, which are later used during the validation process.

The attributes and their associated components are defined as follows:

- Logical Network Sector (SEC): Identifies the logical segment to which the node belongs.
- Installer Signature (INS): Represents the approval of a trusted installer; this attribute is associated with both the network sector and the installer's identity.
- Power Consumption (POC): Describes the typical power usage profile of the node within its sector.
- Transmission Pattern (TRA): Defines the expected communication behavior of the node, again within the context of its assigned network sector.

6.3.3.2 Implementation Details

The implementation specifics are as follows:

- **Attribute Settings:** Attributes for the nodes were set as SEC, INS, POC, and TRA.
- **Hashing Process:** During the generation, the SHA-1 hash function has been utilized.
- **Token Issuance in Registration Phase:** In the registration phase, each Local Node (LN) received four tokens from the MN, corresponding to the four attributes (SEC, INS, POC, and TRA), which were later used during the validation phase.
- **Performance Metrics:** Key metrics, such as time taken for registration, generation, and validation phases, were measured to assess the efficiency of the system (Table 6.1).

Table 6.1: Implementation Time.

Phases	No. of Devices	Time (ms)
Registration	7 LNs	138
Generation	1 LN	0.65
Validation – Challenge	1 LN	38
Validation – Response	1 LN	29

Table 6.2: Comparative Analysis of Consensus Mechanisms.

Mechanism	Throughput	Block Time	Energy	Security
POC	Low	∼10 min	High	Very High
PoS	Med-High	Variable	Low	Med-High
PBFT	High	Seconds	Mod	Low-High
Our Proposal	High	Seconds	Mod	High / ABS

6.3.4 Security Analysis Justification

The proposed security framework addresses critical vulnerabilities inherent in IIoT environments, such as privacy leakage, security flaws, and resource constraints. It incorporates advanced encryption and robust key management mechanisms to safeguard user identities and transaction integrity, minimizing risks associated with unauthorized data exposure and ensuring strong protection against privacy breaches.

Additionally, the framework's resilience against both passive and active adversarial threats, such as data interception and DoS attacks, is strengthened by employing redundancy and resilient architectural designs. These proactive measures ensure continuous network functionality, even amidst targeted disruptions or node failures, thereby significantly enhancing the operational robustness and reliability of the IIoT ecosystem.

6.3.4.1 Performance Analysis

Consensus mechanisms play a crucial role in balancing transaction speed, energy consumption, and system security in IIoT applications. We conducted a comparative study of PoW [237], Proof of Stake (PoS) [238], and PBFT [239]. Our proposed system, optimized for IIoT contexts, redefines the role of resource-limited local nodes, allowing them to safely generate data while not being directly involved in the consensus process.

The proposed system improves efficiency (Table. 6.2) by redefining the function of resource-limited local nodes, enhancing network security even if not directly involved in the consensus process. Full nodes create blocks by consensus, including stored information from local nodes. This system accommodates resource constraints in IIoT contexts while maintaining high throughput and low latency features of PBFT.

The proposed architecture was analyzed for its performance, focusing on consensus mechanisms. The approach uses the PBFT model, enhancing candidate block formation by MNs, who pre-process higher-level information, which is then consoli-

dated by FNs during block creation.

The modification in the design ensures that candidate blocks by FNs have already undergone a preliminary verification and secure storage by MNs, making the process more secure and efficient. This design is particularly beneficial in IIoT contexts, where resource constraints are common. It maintains the high throughput and low latency characteristics of traditional PBFT while optimizing it for IIoT environments, ensuring swift and secure transaction validation.

Shopfloor blockchain solution offer secure and privacy-preserving industrial operations, but their applicability is limited by computational overhead and transaction latency. Industrial environments require efficiency in real-time decision-making, especially when integrating blockchain at the IIoT device level. To achieve high transaction throughput with minimal resource consumption, a lightweight blockchain architecture is needed. This lightweight approach allows IIoT devices to participate in blockchain networks without excessive computational and storage burdens, enabling practical and scalable deployment across industrial settings.

6.4 Lightweight Blockchain Approach

While the proposed shopfloor blockchain architecture effectively addresses scalability, latency, and data processing inefficiencies in IIoT, it still poses challenges regarding resource consumption and processing overhead, especially for highly constrained industrial devices. To overcome these limitations, this section introduces a complementary lightweight blockchain approach. This new method enhances processing speed by simplifying block validation and employing a streamlined attribute verification mechanism. By integrating this lightweight approach with the previously described architecture, the overall system achieves a balanced trade-off—combining the security and fault tolerance benefits of the robust shopfloor solution with the efficiency and rapid processing required for resource-limited industrial environments.

The design of the proposed architecture aims to address the limitations of current blockchain solutions in IIoT settings, such as inability to scale, high latency, and inefficiency in processing large amounts of data generated by industrial devices. The proposed architecture combines the strengths of BFT and DAG, providing security through attribute base verification and fault tolerance, while also offering weak anonymity and secrecy by design.

- **Weak Anonymity:** The Transaction Verification Nodes (TVN) can perform the verification process without knowing the generated data or the node's ID, except when the ID is sent as part of the verification process.
- **Secrecy:** The verification process does not involve Committee Nodes (CN) in token usage by regular nodes or public key usage by transaction verification nodes, including their usage and timing.

6.4.1 Network Architecture and Node Roles

Figure 6.4 illustrates our BFT-DAG architecture using attribute-based verification. IIoT nodes work together seamlessly, using advanced protocols for efficient communication and transaction processing, which is crucial for maintaining network

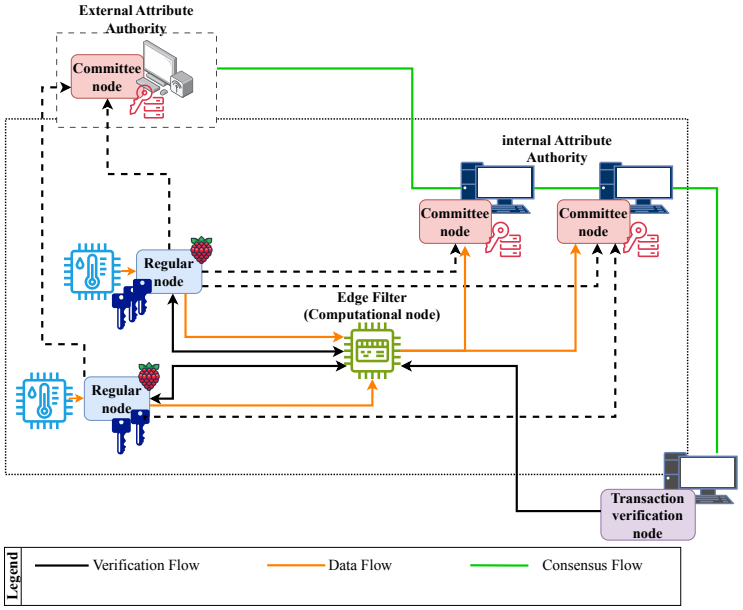


Figure 6.4: Relationship Between Main Players in the Proposed Architecture.

integrity, security, and performance. The BFT-DAG architecture comprises a network of nodes categorized into four types:

Regular Nodes (RNs): IIoT nodes, such as sensors and actuators, support the network by creating validated data through edge computing nodes. They have lower computational power and focus on data collection, generation, or actuation within the IIoT environment. They play a role in the Prover role in attribute verification protocol.

Edge Computing Nodes (ECNs): Data pre-processing is crucial for nodes close to data sources, handling tasks like data aggregation, filtration, and preliminary analysis. These nodes reduce data volume for transmission and validation. They minimize latency by undertaking computational tasks, alleviating load on core network components. They create transactions and send them to the mempool, aligning with the Prover role, where they aggregate and process attributes.

Committee Node (CN): Committee nodes are essential in maintaining the ledger, verifying transactions, and carrying out the consensus mechanism in the DAG. They ensure transactions are logically and securely linked, automatically sorting them without explicit commit steps. This technique offers a scalable and effective consensus mechanism suitable for changing network conditions. During transaction validation, committee nodes perform powerful computations that may not be executable on regular nodes, increasing throughput and decreasing latency.

Transaction Verification Nodess (TVNs): In the BFT-DAG system, nodes revalidate transactions to ensure only valid nodes with valid tokens create transactions, maintaining network security and integrity. This mirrors the Verifier

role in attribute verification protocol, where the verifier challenges provers to present valid tokens.

The network can have multiple subnetworks, each with 1 to n regular nodes, 1 to $n-1$ edge computing nodes, at least 1 transaction verification node, and 0 to n committee nodes. However, the total number of committee nodes in the network must be at least 3 for reliable consensus.

6.4.1.1 Initialization Process

The initialization process is crucial for setting up the system, authenticating nodes, and establishing the primary node for the committee. This process is detailed in Algorithm 14 and involves several clearly defined steps:

Algorithm 14 starts by randomly selecting a primary node from the pool of CNs (line 2). The selected primary node is then verified by other Committee Nodes to ensure it meets required security and performance criteria (lines 3–6). If the node fails verification, the selection process is repeated until a valid primary node is identified.

Following this, the algorithm proceeds to authenticate nodes within each subnetwork (lines 8–18). Committee Nodes authenticate RNs (lines 9–11) and ECNs (lines 12–14), verifying their attributes and confirming their validity and authorization. Additionally, mutual authentication between ECNs and RNs is conducted to establish secure communication channels within subnetworks (lines 15–18).

Finally, once the primary node has been successfully verified and all subnetworks have completed node authentication, the algorithm finalizes the network initialization (lines 20–22). With all nodes authenticated, the blockchain network becomes fully operational, enabling the system to begin normal transaction processing and consensus mechanisms.

6.4.1.2 Transaction Creation and Validation Process

The system initiates the network and authenticates nodes before proceeding with transaction creation and validation, detailing each step and its detailed algorithm. Algorithm 15 proceeds as following:

1. **Transaction Generation (Lines 1-7):** RNs collect data from sensors or actuators, format it into a transaction with a timestamp and unique identifier, and sign it with a private key for authenticity and integrity, which is then sent to an ECN.

2. **Data Pre-Processing (Lines 8-13):** ECNs receive transactions from RNs, aggregate them, filter out redundant or irrelevant data, and perform a preliminary analysis before adding processed transactions to the mempool for validation.

3. **Transaction Validation (Lines 14-21):** TVNs verify transactions from mempools using attribute tokens to ensure data validity and sender authenticity. Once validated, transactions are forwarded to CN for inclusion in the DAG, ensuring only legitimate transactions are added to the blockchain, thereby maintaining the network's integrity and security.

Algorithm 14: Network Initialization

Input: List of CNs, RNs, ECNs
Output: Initialized blockchain network

```
// Step 0: Genesis bootstrap
Preselect an initial set of Committee Nodes (CNs);
Issue genesis X.509 certificates τ_CN^(0) to these CNs;
// Step 1: Candidate CN joins
Existing CNs jointly issue certificate τ_CN = g^α_i H(ID_CN)^β_i to candidate CN;
Candidate CN broadcasts τ_CN to all CNs;
// Step 2: AVP Proof for candidate CN
All CNs perform n-out-of-n Attribute Verifier Protocol (AVP) proof on τ_CN;
if AVP fails for any CN then
    Quarantine the candidate CN;
// Step 3: Select and verify the primary node
primary_node ← Randomly select a node from CN;
if Verify(primary_node) = TRUE then
    Set primary_node as the primary committee node;
else
    Repeat step until a valid primary_node is found;
// Step 4: Authenticate nodes within each subnetwork
foreach CN in the network do
    foreach RN in CN.subnetwork do
        if Verify(RN) = TRUE then
            Authenticate RN;
    foreach ECN in CN.subnetwork do
        if Verify(EN) = TRUE then
            Authenticate ECN;
        // Mutual authentication between ECN and RNs
        foreach RN in EN.subnetwork do
            if MutualAuthenticate(ECN, RN) = TRUE then
                Establish secure channel between ECN and RN;
// Step 5: Finalize initialization
if All nodes authenticated successfully then
    Initialize network;
    Start normal transaction processing;
```

6.4.1.3 Consensus and Inclusion in DAG

The proposed lightweight blockchain architecture utilizes a consensus mechanism to ensure the integrity, security, and consistency of the distributed ledger. This mechanism involves CN executing a BFT protocol to validate and order transactions, providing a detailed explanation of transaction inclusion.

1. **Transaction Propagation:** A transaction is generated and pre-processed, then propagated to the mempool for validation by the TVN for authenticity and attribute validity before being picked up by Committee Nodes.

2. **Proposal Phase:** A CN, often the primary node, collects transactions from

Algorithm 15: Transaction Creation and Validation

Input: Data from RN, ECN, TVN
Output: Validated transactions ready for inclusion in the DAG

```
// Step 1: Transaction Generation
foreach RN do
    data ← Collect data from sensors/actuators;
    transaction ← Create transaction with data, timestamp, and unique identifier;
    signed_transaction ← Sign transaction with RN private key;
    Send signed_transaction to ECN;
// Step 2: Data Pre-Processing
foreach ECN do
    transactions ← Receive transactions from RNs;
    aggregated_data ← Aggregate transactions;
    filtered_data ← Filter redundant/irrelevant data from aggregated_data;
    pre_processed_transactions ← Perform preliminary analysis on filtered_data;
    Add pre_processed_transactions to mempool;
// Step 3: Transaction Validation
foreach TVN do
    transactions ← Retrieve transactions from mempool;
    foreach transaction in transactions do
        if Validate(transaction) = TRUE then
            if VerifyAttributeTokens(transaction) = TRUE then
                validated_transaction ← transaction;
                Send validated_transaction to CN;
```

the mempool and creates a proposal block, including these transactions and references to previous vertices.

3. **Voting Phase:** The proposal block is broadcasted to all other CNs, who independently verify the transactions against the current state of the DAG, checking transaction validity, preventing double-spending, and confirming protocol rules adherence.

4. **Pre-Commit Phase:** After verification, each CN sends a pre-commit vote to other CNs, which is aggregated to determine if a quorum (typically 2/3 majority) agrees on the proposal block.

5. **Commit Phase:** The proposal block is committed if all CNs receive enough pre-commit votes, and each CN sends a commit vote, which is aggregated to ensure consensus, and a quorum is reached.

6. **Finalization:** The committed block is added to the DAG, linking it to previous vertices, and the updated state is broadcasted to all network nodes, updating their local ledgers.

Algorithm 16 begins with a proposal phase where a node from the CNs creates a block with mempool transactions and references to previous vertices. In the voting phase, each Committee Node verifies transactions against the current state of the

Algorithm 16: BFT-DAG Consensus Mechanism

Input: Transactions from mempool
Output: Updated DAG with committed transactions

```
// Step 1: Proposal Phase
proposer ← Select a Committee Node to propose a block;
proposal_block ← Create block with transactions from mempool;
proposal_block.references ← References to previous DAG vertices;
Broadcast proposal_block to all Committee Nodes;
// Step 2: Voting Phase
foreach CN in Committee Nodes do
    if Verify(proposal_block) = TRUE then
        pre_commit_vote ← Generate pre-commit vote;
        Broadcast pre_commit_vote to all Committee Nodes;
// Step 3: Pre-Commit Phase
pre_commit_votes ← Collect pre-commit votes;
if Quorum(pre_commit_votes) = TRUE then
    foreach CN in Committee Nodes do
        commit_vote ← Generate commit vote;
        Broadcast commit vote to all Committee Nodes;
// Step 4: Commit Phase
commit_votes ← Collect commit votes;
if Quorum(commit_votes) = TRUE then
    Add proposal_block to DAG;
    Broadcast updated DAG to all nodes;
```

DAG to ensure transaction validity and protocol rules adherence. In the pre-commit phase, eachCN sends a commit vote, which is aggregated and considered committed if a quorum is reached. The committed block is added to the DAG, and the updated state is broadcasted to all nodes in the network.

Inclusion in DAG

1. **Vertex Creation:** Each committed block creates a new vertex in the DAG, referencing previous vertices, ensuring the chronological order of transactions.

2. **Linking Vertices:** Cryptographic hashes link new vertex transactions to previous ones, preventing tampering and making the cryptographic chain immediately detectable.

3. **Updating the Ledger:** Once a new vertex is added to the DAG, all nodes update their local copies of the ledger. This update includes the latest state of the DAG, ensuring consistency across the network.

The consensus method, depicted in Figure 6.5, demonstrates the interconnected roles of different node types and their collaboration, illustrating the operational flow from transaction generation to validation.

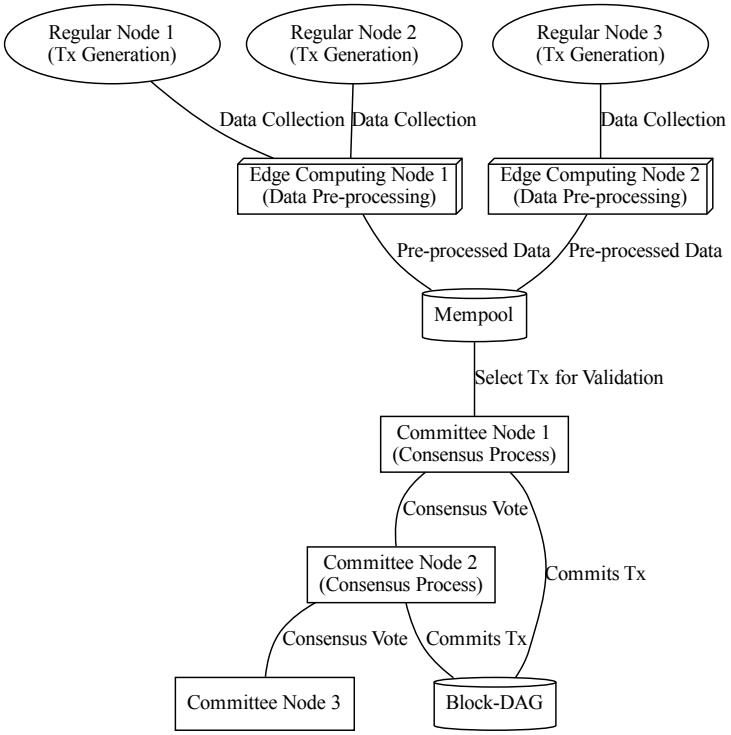


Figure 6.5: Data Flow and Consensus Process Among Different Nodes Within a BFT-DAG System.

6.4.2 Security Analysis

The BFT-DAG architecture focuses on security to protect IIoT systems from vulnerabilities. It addresses Replay, DoS, and Man-in-the-Middle (MitM) attacks, but also considers broader models like node tampering, data forgery, Sybil attacks, Distributed Denial-of-Service (DDoS), and side-channel attacks. This comprehensive approach ensures a robust evaluation of the architecture's resilience.

6.4.2.1 Addressed Threats

Replay Attack

During the verification process, the TVN generates a challenge as follows:

$$challenge_{TVN} = \text{Enc}(r||ts||PK_{TVN}) \tag{6.12}$$

where r is a random parameter, ts is the timestamp, and PK_{TVN} is the public key of the corresponding transaction verification node. By including both a nonce value r and the current timestamp ts, and assuming a cryptographically secure random number generator and synchronized system time, replayed verification packets are directly detected.

Table 6.3: Comparison of Mitigation Strategies Against Common Attacks.

Attack Type	Impact	Mitigation	Evaluation
Replay	Data integrity	Nonce, timestamp validation	Mitigated
DoS	Network availability	Decentralized design	Mitigated
MITM	Data confidentiality	Encrypted communication	Mitigated
Data Forgery	Data authenticity	Cryptographic signatures	Mitigated
DDoS	Network availability	Distributed processing	Partially mitigated
Sybil	Consensus reliability	Attribute verification	Partially mitigated
Side-Channel	Confidentiality	Secure hardware, randomization	Partially mitigated

Denial of Service (DoS) Attack

The system's decentralized design ensures that no single managing entity exists. During the transaction verification process, external third-party nodes are not involved, including issuing committee nodes. Consequently, a DoS attack targeting the transaction verification process is ineffective, as there is no central point of failure.

Man-in-the-Middle (MITM) Attack

Committee nodes generate tokens for each RN based on their attributes:

$$token_{(RN,attribute)} = PK_1 \cdot I^{PK_2} \tag{6.13}$$

where PK_1 and PK_2 are the public keys of the issuing committee nodes, and I is the identifier of the RN. With encrypted traffic during transaction creation and verification, unauthorized interception or manipulation is effectively mitigated.

There are some additional vulnerabilities to address such as Data forgery, which involves the malicious creation or modification of transactions, is mitigated by employing cryptographic signatures to ensure authenticity and using attribute-based verification to restrict transaction creation to authorized nodes.

Similarly, DDoS attacks designed to overwhelm network resources are countered through decentralized processing and consensus mechanisms that eliminate single points of failure, with resource-intensive tasks delegated to Committee Nodes capable of handling high loads.

The architecture also addresses Sybil attacks by using attribute-based verification to authenticate nodes before participation and by incorporating a reputation system to penalize malicious behavior. In addition, side-channel attacks, which exploit unintended information leakage such as timing or power consumption, are mitigated by randomizing computational processes to obscure timing patterns and by leveraging secure hardware to minimize such leakage.

Table 6.3 summarizes the architecture's resilience against various attacks.

6.4.3 Implementation and Evaluation

This section outlines the methodology for implementing the proposed architecture, which involves combining hardware and simulated network conditions to simulate an IIoT environment.

6.4.3.1 Implementation Methodology

The proposed BFT-DAG architecture was evaluated through a comprehensive hardware implementation, allowing for a thorough exploration of its response to various network scenarios. Our hardware setup is illustrated in Figure 6.6.

Figure 6.6: Hardware Network Demonstration for Test Architecture.

Three laptops serve as primary nodes or committee members in the BFT-DAG network, responsible for proposing, validating, and committing transactions. One laptop is designated as an ECN, responsible for data aggregation and computational processing near data sources, reducing latency and offloading processing tasks from the central network. The laptops communicate through a local network setup with WLAN-Standard IEEE 802.11n (Wi-Fi 4), effectively simulating the network interactions typical in IIoT deployments.

The Raspberry Pi acts as a gateway, enabling seamless communication between environmental sensors and edge computing nodes. It manages the digital temperature sensor (TMP102) and analog light sensor (MCP3008 ADC) via I2C and SPI protocols, ensuring efficient pre-processing and transmission of real-time data for further action. This ensures efficient data collection and processing. Table 6.4 regroups the specification of our hardware platform.

6.4.3.2 Results of the Evaluation

We measured performance evaluation of the architecture based on three key metrics: ***verification time***, ***throughput***, and ***latency***. Verification time measures the duration required to validate a transaction's authenticity and integrity including signature validation and data consistency checks with the current ledger state. Throughput quantifies the number of Transactions Processed per Second (TPS), reflecting the network's capacity to handle high volumes of transactions, while latency captures the time elapsed from transaction initiation to its final confirmation and inclusion in the blockchain ledger, thereby indicating the overall speed and efficiency of the system.

Table 6.4: Hardware Information for Testing the BFT-DAG Architecture.

Device Model	Network Role	Specifications
Lenovo IdeaPad 520	CN	Intel i7 8550U, 16GB RAM
Lenovo ThinkPad x230	CN	Intel i3 3110M, 8GB RAM
Lenovo IdeaPad Flex 15	CN / ECN	Intel i3 4010U, 4GB RAM
RP2040	Gateway	Dual-core ARM Cortex M0+ processor, 133 MHz, 264KB of SRAM, 2MB on-board flash memory
Light Sensor GL5528	Data Collector	Light Intensity Measurement
Temperature Sensor DHT22	Data Collector	Temperature and Humidity Measurement

Table 6.5: Verification Time Across Different Devices.

Component	Time (ms)
Prover for Response Creation Task (RN)	13.013
Verifier for Verification Task (CN)	10.74
Verifier for Lightweight Protocol Task (CN)	3.142

Verification Time

The verification time was evaluated on three laptops (Laptop 1, Laptop 2/Raspberry Pi, and Laptop 3), focusing on the prover's response generation time and the verifier's verification time. Table 6.5 summarizes these results.

The verification time of three devices was evaluated, focusing on the time taken by the prover and verifier in different tasks. The prover, typically a RN, takes the longest at 13.013 milliseconds due to cryptographic proof generation complexity. The verifier, a CN, takes 10.74 milliseconds for standard verification. The lightweight protocol verification is the quickest at 3.142 milliseconds, indicating efficiency with fewer cryptographic checks or simpler algorithms.

Energy Consumption Evaluation

This section evaluates the energy efficiency of the proposed BFT-DAG architecture using hardware specifications and throughput results from experiments. Energy consumption is calculated based on power consumption of each node type and the system's TPS. Using the formula:

$$\text{Energy per transaction (J)} = \frac{\text{Power (W)}}{\text{TPS}}$$

The study estimated energy usage for RNs, ECNs, and CNs, including authentication overhead. The total energy per transaction was calculated, summarizing energy consumption for each node type in Table 6.6.

The system's total energy consumption per second, including authentication overhead, is 55.05 Joules for a transaction throughput of 49,039 TPS, indicating its

Table 6.6: Energy Consumption Per Transaction and Total Energy Per Second.

Node Type	**Power (W)**	**Base Energy/-Transaction (J)**	**Authentication Overhead (J)**	**Total Energy/-Transaction (J)**
RN	2.5	0.000051	0.00000255	0.00005355
ECN	20	0.000408	0.0000204	0.0004284
CN	30	0.000612	0.0000306	0.0006426
Total	52.5 W	0.00107 J	0.00005355 J	0.00112355 J

Table 6.7: Benchmarking Results for Consensus Throughput and Latency.

Metric	**Result**
Consensus TPS (Transactions Per Second)	49,439 tx/s
Consensus BPS (Bytes Per Second)	25,312,723 B/s
Consensus Latency	433 ms
End-to-End TPS	49,039 tx/s
End-to-End BPS	25,107,962 B/s
End-to-End Latency	577 ms

suitability for energy-efficient, high-throughput environments.

6.4.3.3 Benchmarking Results

The benchmark evaluates consensus throughput and latency across various transactions and batch sizes. The consensus throughput achieves 49K TPS and 25 Mbps, with a latency of 433 milliseconds, indicating high efficiency in transaction processing within the consensus mechanism. The end-to-end system, including transaction initiation to finalization, shows nearly the same throughput with a slight decrease of less than 1%. Latency slightly increases to 577 milliseconds, attributed to cumulative processing and network delays beyond the consensus layer. These results demonstrate the system's ability to handle high transaction volumes efficiently and suggest areas for further optimization for improved performance.

6.4.3.4 Comparison With Related Works

The proposed architecture's performance evaluation involves three comparisons: average TPS, consensus latency, and the balance between TPS and latency. These metrics reveal the system's strengths in rapid data processing and accurate consensus. Verification and benchmarking tests confirm its high throughput and efficient consensus capabilities, making it suitable for distributed environments requiring reliable and low-latency consensus.

Figure 6.7 compares average TPS across several blockchain systems, revealing the efficiency of the proposed architecture in high-throughput scenarios. Bitcoin, known for its PoW mechanism, has high-energy consumption and slow transaction processing, resulting in limited throughput. Ethereum, currently transitioning from PoW to PoS with Ethereum 2.0, offers moderate throughput. Solana combines Proof of History (PoH) with PoS, providing higher throughput and lower latency compared to PoW systems. GradedDAG and Shoal++ excel in processing high

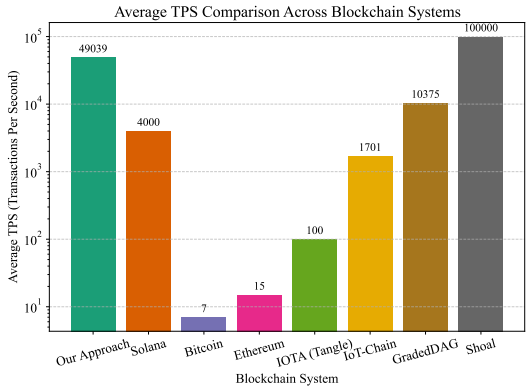


Figure 6.7: Average TPS Comparison Across Blockchain Systems.

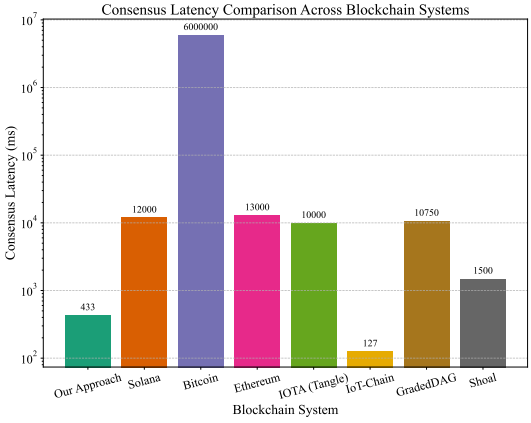


Figure 6.8: Consensus Latency Comparison Across Blockchain Systems.

volumes of transactions with reduced latency, enhancing throughput significantly over traditional blockchain systems.

IOTA's Tangle uses a DAG instead of a traditional blockchain, making it better suited for IoT applications with lower transaction fees and improved scalability. IoT-Chain, designed specifically for IoT environments, focuses on low latency and efficient consensus. Our architecture after Shoal achieves significantly higher TPS compared to its competitors.

While throughput is crucial, achieving consensus promptly is equally important. Figure 6.8 compares consensus latency across blockchain systems. The proposed architecture's latency of 433 milliseconds is markedly lower than that of systems like Bitcoin, which suffer from long delays due to their PoW protocols. Bitcoin's approximately 10-minute latency range highlights the inefficiencies of these systems for applications requiring prompt decision-making. Ethereum's latency is 12–15 seconds, Solana, GradedDAG, Shoal++, and IOTA's Tangle show better latency but are still not as efficient as our system. IoT-Chain achieves lower latency due to its design for IoT environments, but our architecture strikes a better balance between high throughput and low latency, making it ideal for real-time applications.

Figure 6.9 illustrates the relationship between TPS and consensus latency. Our

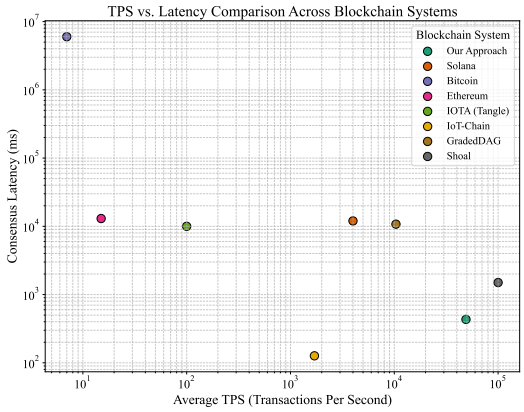


Figure 6.9: TPS vs. Latency Comparison Across Blockchain Systems.

Table 6.8: Comparison of Energy Consumption Across Blockchain Systems.

System	**Energy/Transaction (J)**	**Energy/Second (J)**
Proposed BFT-DAG	0.00112355	055.05
Solana	0.01483200	740.82
Bitcoin	2,545,200	N/A
Ethereum (PoW)	225,000	N/A
Ethereum (PoS)	0.010800	N/A
IOTA	0.000001	N/A

system demonstrates a compelling balance, maintaining high throughput while ensuring relatively low latency. Bitcoin achieves lower TPS with high latency due to its PoW mechanisms. Ethereum, Solana, GradedDAG and IOTA's Tangle improve on this but still face trade-offs. Our architecture, optimized for both metrics, offers a significant advantage for applications demanding rapid data processing and reliable, low-latency consensus. This balance is essential for IoT applications, where quick data processing and immediate consensus are critical for performance and reliability.

To provide context, we compare the estimated energy consumption of our system with other blockchain systems, including Solana, Bitcoin, Ethereum, and IOTA (Table 6.8).

Our system demonstrates superior energy efficiency compared to traditional blockchain systems like Bitcoin and Ethereum (PoW). Although platforms like Solana and IOTA also exhibit low-energy consumption, the proposed BFT-DAG balances energy efficiency, high throughput, and robust security, making it particularly suitable for IIoT environments.

6.5 Conclusion

This chapter introduced two original blockchain-based contributions aimed at significantly enhancing security in distributed IIoT environments. These solutions specifically address critical issues of data integrity, privacy, scalability, and efficient resource management within complex industrial settings.

The first contribution, the "Shopfloor Blockchain Approach," provides an original, secure, and privacy-preserving blockchain architecture uniquely tailored for industrial auditing applications. Its novelty lies in integrating blockchain with advanced privacy-preserving mechanisms, directly addressing traditional auditing limitations such as secure real-time data management, verifiable audit trails, and confidentiality across geographically distributed shopfloor operations. The tailored design ensures that sensitive industrial data remains secure and traceable, specifically benefiting audit-intensive environments requiring rigorous compliance and transparency.

The second contribution, the "Lightweight Blockchain Approach," presents an original blockchain model specifically optimized for resource-constrained IIoT devices, an essential requirement overlooked by traditional blockchain implementations. Its distinctive feature is the significant reduction in computational complexity combined with an efficient consensus mechanism designed explicitly for industrial contexts with limited processing power. This specialized approach markedly improves transaction throughput and latency, ensuring the viability and practicality of blockchain technology even on devices with constrained resources.

Despite these advancements, some limitations persist. The Shopfloor Blockchain Approach relies on specialized hardware, limiting its applicability in scenarios demanding stringent real-time timing. Meanwhile, the Lightweight Blockchain Approach significantly improves scalability and efficiency but faces challenges in maintaining strong anonymity and secrecy under highly dynamic network conditions, indicating opportunities for further enhancements.

In conclusion, these innovative blockchain-based frameworks meaningfully advance security for distributed industrial environments by addressing specific operational constraints and data management requirements. Building upon these novel foundations, the next chapter explores their integration into comprehensive, dynamic security frameworks, further progressing toward a cohesive and adaptive security ecosystem for IIoT.

Chapter 7

Dynamic Zero Trust Architecture

Contents

Publications:

- Fatemeh Stodt, Christoph Reich, and Fabrice Theoleyre. “Beyond Static Security: A Context-Aware and Real-Time Dynamic Zero Trust Architecture for IIoT Access Control”. In: ***IEEE Internet of Things Journal*** (2025)

7.1 Introduction

Traditional Zero Trust Architectures (ZTAs) have played a key role in modern cybersecurity by enforcing the principle of "never trust, always verify" [240]. However, their static and predefined policy frameworks are increasingly inadequate in IIoT environments, where security requirements change rapidly due to evolving device interactions, operational states, and external threat conditions.

In dynamic and distributed industrial systems, access decisions must adapt continuously to contextual information such as user behavior, device status, network conditions, and real-time threat intelligence. Static ZTA implementations, while robust in controlled enterprise settings, often fail to reflect this situational awareness, resulting in delayed responses, excessive privilege grants, or unnecessary access denials.

To overcome these limitations, this chapter introduces a novel contribution that integrates contextual intelligence into Zero Trust enforcement:

Dynamic Zero Trust Architecture (Dynamic ZTA)(Section 7.2): This framework enables adaptive access control by continuously evaluating contextual data and dynamically adjusting access permissions in real time. Unlike prior ZTA models, such as the static-policy-based architecture proposed by Paul and Rao [74], our approach incorporates live context signals and continuous risk assessments to refine trust decisions on the fly. This dynamic mechanism significantly enhances responsiveness and reduces security blind spots caused by rigid policy enforcement.

The originality of this contribution lies in its combination of Zero Trust principles with real-time context evaluation and threat-aware decision-making. This ensures that IIoT systems maintain both fine-grained access control and high adaptability, even in the face of changing conditions and adversarial activity. The proposed architecture not only addresses key security gaps in traditional ZTA, but also serves as a foundation for the more comprehensive distributed and blockchain-integrated security solutions discussed in subsequent chapters.

Addresses Research Questions:

- **What are the limitations of traditional Zero Trust architectures in dynamic IIoT environments, and how can they be adapted for real-time security?**

7.2 Dynamic Zero Trust Framework Overview

The dynamic ZTA design addresses complex security issues in network environments by focusing on adaptive verification, continuous risk assessment, and dynamic policy enforcement to ensure trust is dynamic and context-validated.

Our framework is unique in addressing contextual information and network segment criticality, unlike existing solutions that focus on specific aspects like user behavior or device compliance. By integrating threat assessment with continuous policy adaptation, we achieve a more granular and proactive security model, making it a scalable and efficient solution for heterogeneous and evolving environments like the IIoT.

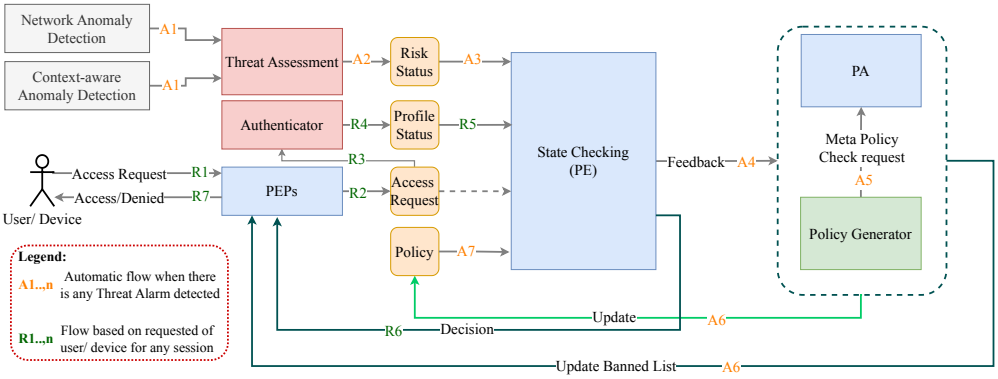


Figure 7.1: Integrated Framework for Dynamic Zero Trust Architecture.

The system operates as an interconnected framework as illustrated in Figure 7.1, where each component collaborates to form a robust security posture. This architecture ensures that every access request is carefully considered and managed dynamically based on the context of the environment.

7.3 Architectural Components

7.3.1 Core Properties

We have designed our solution to provide the following characteristics:

- **Dynamic Trust Adjustments:** trust within the network is dynamic and varies based on context, device compliance, and user behavior. The architecture continuously assesses risk levels to adjust access permissions in real-time, ensuring security responses are aligned with current conditions.
- **Continuous Policy Adaptation:** because static security policies are inadequate in a dynamic IIoT environment, our policy generator continuously updates policies based on real-time assessments, ensuring the architecture remains responsive to emerging risks.
- **Core Infrastructure Integrity Assumption:** the architecture assumes core components are initially uncompromised, enabling reliable anomaly detec-

tion. Any deviations are flagged, allowing the Policy Engine to adjust access permissions proactively.

Identifiers and Access Workflow

The operational workflow begins when an access request is made by an entity (such as a user, device, or application) seeking to access a network resource. Each entity is assigned a unique identifier, referred to as an *ID*, which the system uses to manage and authenticate access permissions. In this architecture:

- ***User ID*** identifies an individual or a group.
- ***Device ID*** identifies a device within the network.
- ***Flow ID*** Flow ID refers to a unique identifier for a network communication session, typically derived from attributes such as source IP address, destination IP address, source port, destination port, and protocol (5-tuple). It identifies a data flow or session, especially useful for monitoring continuous data streams.

IDs are generated and distributed centrally by the Policy Administration module during device onboarding or user registration. Device IDs are automatically distributed upon device enrollment, user IDs are provisioned during account creation, and flow IDs are generated dynamically per communication session by network monitoring components, then shared with the Policy Enforcement Point (PEP) and Policy Engine (PE) for consistent reference throughout access control processes.

The *ID* plays a central role in the access control workflow. The Policy Enforcement Points (PEP) verifies the entity's ID against a list of authorized or restricted entities when it receives an access request. In that way, only approved IDs can proceed to the authentication stage. The *Authenticator* module then validates the entity's credentials, using the ID to confirm its identity and integrity. Verified IDs are subsequently evaluated by the Policy Engine (PE) based on current security policies, threat assessments, and contextual information to determine whether access should be granted or denied.

Throughout the access request process, the *ID* serves as a consistent reference for tracking, authentication, and decision-making, enabling the system to enforce Zero Trust principles by verifying each unique user, device, and data flow within the network.

Policy Enforcement Points

PEPs are strategically positioned gatekeepers that manage access at critical network intersections [76]. The PEP plays an important role in processing access requests from various entities. The PEP holds a list of banned IDs provided by the Policy Administration (PA). When an access request is received, the PEP checks the requesting entity's ID against this list. If the ID is not present in the banned list, the PEP forwards the entity ID to the Authenticator and the PE. After receiving a decision from the PE, which determines whether to grant or deny access, the PEP completes the access process and logs all attempts and outcomes for audit purposes.

The banned ID list is centrally maintained by the Policy Administration (PA) and synchronized periodically or triggered by security events across all PEPs. IDs can be removed (un-banned) either through administrative intervention after a re-assessment of risk or automatically when anomaly detection indicates normal behavior has resumed. The list is synchronized frequently—typically event-driven—to ensure consistency across all PEPs.

Authenticator

The authenticator validates the identity of users or devices by checking credentials, certificates, and signatures to ensure authenticity. The resulting profile status (valid or not valid) is essential for the PE's decision-making and depends on factors such as certificate validation time, installer signature verification, and approval signature confirmation.

Anomaly Detection

Network Anomaly Detection monitors network traffic to detect deviations from established patterns, indicating possibly a threat. We rely on our previous work [7] for this component. Context-aware Anomaly Detection assesses user and device behavior against historical norms and context of environment such as time, location, and user activities [241].

Policy Generator

This component updates or creates new policies based on findings from the Threat Assessment module and the security status of entities within the system, ensuring an adaptable security posture. The sum of these interactions guides the decision of the PE to grant or deny access. If the decision is to deny access, the PEP enforces it.

Policy Engine (PE)

The PE evaluates access requests based on real-time security data and adaptive measures. It relies on a Finite State Machine (FSM) to manage and transition between logical states that represent the risk levels of entities. This integration provides a structured approach to handling the dynamic nature of security management, enabling the PE to respond effectively to changes in an entity's behavior.

Within the PE, the FSM (Figure 7.2) defines states which each corresponding to a specific security posture. Entities transition between these states based on triggers generated by real-time threat assessments and contextual analysis. FSM states include Normal, Alert, High Risk, Quarantined, and Compromised, transitioning based on explicit triggers with specific security implications:

- **Normal to Alert**: Triggered by minor anomalies (Threat Risk score between 0.4-0.6), the Threat Risk Score is a quantitative metric aggregating normalized contextual criteria to assess the current security risk associated with a device, user, or network segment. , prompting increased monitoring.

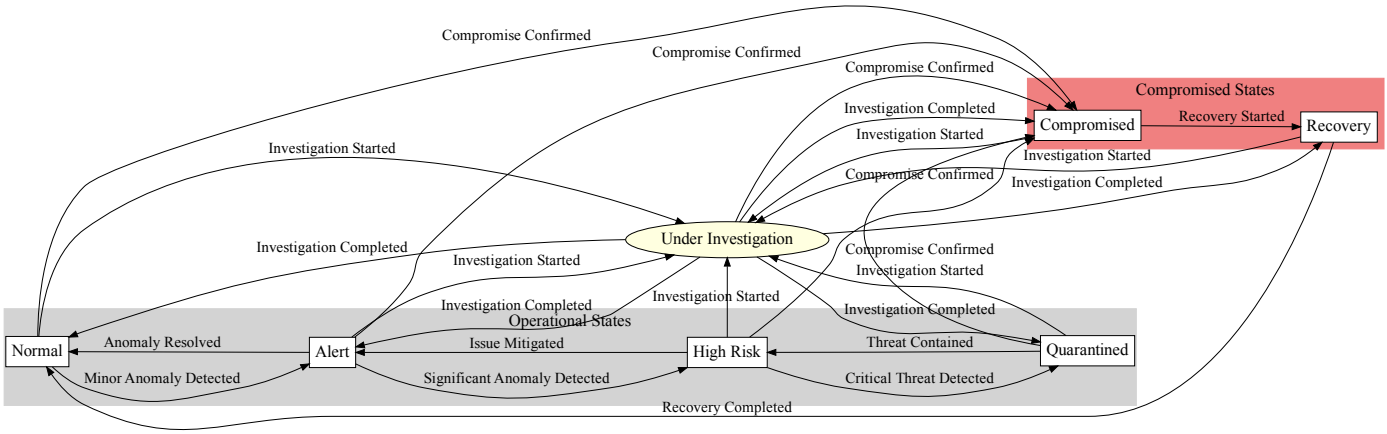


Figure 7.2: Finite State Machine for Policy Engine.

- **Alert to High Risk**: Significant anomalies detected (Threat Risk between 0.6-0.8), triggering heightened security measures and restricted access.
- **High Risk to Quarantined**: Critical threats detected (Threat Risk ≥ 0.8), enforcing isolation and stringent access limitations.
- **Alert/High Risk/Quarantined to Compromised**: Confirmed security breaches via investigation, mandating immediate isolation and remediation.
- **Compromised to Recovery**: Initiation of recovery procedures post-investigation to restore secure operations.
- **Recovery to Normal**: Post-recovery verification and return to standard operational status upon confirmed resolution.

For instance, detecting privilege escalation with a high confidence level prompts the PE to transition the entity to a High Risk or Quarantined state, invoking stricter access measures or isolation protocols. This structured state management enables the PE to apply policy-driven actions tailored to each risk level, ensuring security responses are proportionate and timely.

The FSM includes a feedback loop, explicitly involving continuous monitoring and dynamic updates to authorized or banned entity IDs. If anomalous behavior is detected, entity IDs may be flagged and transitions between states occur accordingly, resulting in updates to the banned ID lists maintained by the Policy Enforcement Point (PEP) and Policy Administration (PA). Conversely, when an entity previously marked as risky demonstrates normal behavior, it transitions to a lower-risk state, potentially resulting in its ID being removed from banned lists. This dynamic interplay ensures robust security by continuously aligning the access control measures with real-time assessments and evolving contextual information.

The low computational complexity of employing a FSM further supports real-time assessment and effective scalability in dynamic environments.

Threat Assessment

This module evaluates the potential risks associated with each access request within ZTA. This module synthesizes inputs from Network and Context-aware Anomaly Detection to derive an overall threat risk score. The assessment employs a structured approach, incorporating statistical analysis and machine learning models to classify threats and assess their severity. This comprehensive analysis informs the Policy Engine (PE) and contributes to adaptive, context-aware decision-making.

Policy Administration (PA)

The Policy Administration (PA) component plays a central role in the proposed ZTA framework, serving as the authority for managing and maintaining security policies that govern access control across the network. It ensures that these policies are effectively deployed and dynamically updated to respond to an evolving security environment. Key responsibilities of the PA include the creation and management of policies, as well as the enforcement of meta-policies that adapt to changing contextual factors. A meta-policy, in this context, refers to a higher-level template that governs the creation and dynamic adjustment of specific access control rules.

Table 7.1: Summary of Variables Used in Threat Risk Calculation.

Symbol	Description
C	Confidence of Threat
A	Attack Criticality
S	Segment Criticality
P	Past Anomalies
C_{norm}	Normalized Confidence of Threat
A_{norm}	Normalized Attack Criticality
S_{norm}	Normalized Segment Criticality
P_{norm}	Normalized Past Anomalies
w_C	Weight for Confidence of Threat
w_A	Weight for Attack Criticality
w_S	Weight for Segment Criticality
w_P	Weight for Past Anomalies

7.4 Threat Risk Scoring Model

Threat Assessment relies on a ***Threat Risk*** score calculated by combining four key criteria:

1. **Confidence of Threat** (C) represents the model's confidence level in identifying potential threats based on real-time data.
2. **Attack Criticality** (A) is a fixed value assigned based on the severity and potential impact of the detected attack type.

3. **Segment Criticality** (S) represents the importance or sensitivity of the targeted network segment.

4. **Past Anomalies** (P) reflects the historical frequency and severity of anomalies associated with the entity (e.g., user or device).

Each criterion is computed and normalized to ensure consistency across different scales before being combined into the overall Threat Risk score.

7.4.1 Confidence of Threat (C)

C is derived from real-time analysis using machine learning models or statistical methods that detect anomalies or malicious activities. For example, an intrusion detection system may assign a confidence score between 0 and 100% indicating the likelihood that observed behavior is malicious. We apply a Min-Max Scaling for normalization:

$$Var_{\text{norm}} = \frac{Var - Var_{\text{min}}}{Var_{\text{max}} - Var_{\text{min}}}$$

where Var_{min} and Var_{max} are the minimum and maximum possible confidence scores (typically 0 and 100) for a given variable Var, and Var_{norm} is its normalized value.

Min-Max Scaling is the most relevant here because the scores have a fixed range and need to be mapped consistently to [0,1]. This approach preserves the proportionality and interpretability of the data, which is critical for combining multiple criteria in the risk assessment. Alternative methods, like Z-score normalization, are less suitable as they distort the original scale.

7.4.2 Attack Criticality (A)

A is assigned based on the severity of the detected attack type, using standardized threat intelligence sources like the MITRE ATT&CK framework [242]. Each attack type is mapped to a criticality score reflecting its potential impact. Similar to C, A is normalized using Min-Max Scaling.

7.4.3 Segment Criticality (S)

S represents the importance of the network segment being accessed. Segments are assigned criticality scores based on factors like data sensitivity and business impact. Examples of criticality scoring include assigning higher scores to segments containing sensitive operational data (e.g., production control systems) or those directly impacting safety and business continuity. Prior works such as [72] and [75] provide methods for quantifying segment criticality. We still employ Min-Max Scaling for normalization.

7.4.4 Past Anomalies (P)

P quantifies the historical frequency and severity of anomalies linked to the entity requesting access. This includes prior incidents and behavioral deviations.

A logarithmic transformation is applied to manage skewness due to outliers:

$$P_{\log} = \log(P + 1)$$

followed by robust scaling:

$$P_{\text{norm}} = \frac{P_{\log} - \text{Median}(P_{\log})}{\text{IQR}(P_{\log})}$$

where $\text{Median}(P_{\log})$ is the median and $\text{IQR}(P_{\log})$ is the interquartile range of the transformed data.

7.4.5 Threat Risk Calculation and Categorization

The risk assessment process combines anomaly scores obtained from both network anomaly detection and context-aware anomaly detection modules. We compute a unified threat risk score using a weighted linear combination method, ensuring a clear and interpretable representation of the risk level for each entity:

$$\text{Threat Risk} = \sum_{v \in \{C,S,A,P\}} w_v \cdot v_{\text{norm}}, \quad \text{where } \sum w_v = 1 \tag{7.1}$$

Here, the weights w_C, w_A, w_S, and w_P represent the relative significance assigned to the criteria: Confidence of Threat (C), Attack Criticality (A), Segment Criticality (S), and Past Anomalies (P), respectively.

We selected this approach due to its inherent transparency and interpretability, which allows stakeholders to clearly understand the contributions of each risk factor. The weighted combination model offers flexibility to dynamically adjust risk criteria based on evolving threat landscapes and operational priorities within IIoT environments.

Initial weights are expected to be configured by cybersecurity experts or system administrators who possess contextual awareness of the industrial network's operational priorities and the criticality of individual segments. Their domain-specific insight ensures that weighting reflects the relative importance of each risk factor in a real-world deployment.

In our experimental setup, these weights were manually assigned based on predefined assumptions and simulated criticality of entities within the testbed. To reflect realistic prioritization, we employed a contextual risk sensitivity tuning strategy. For example, a simulated sensor managing core pressure in a real-time production machine was treated as highly sensitive due to its immediate safety and operational impact. Conversely, a robotic transporter operating in a non-critical post-assembly stage (e.g., moving finished parts) was considered less sensitive, as it could be temporarily replaced by manual labor without compromising system safety or integrity.

This prioritization was guided by domain-driven assessments rather than statistical tuning, ensuring that the calculated Threat Risk score reflected not only the presence of anomalies but also the potential operational consequences of those anomalies. Currently, weights (wC, wA, wS, wP) are manually configured based on domain expertise, which provides clear interpretability but inherently limits adaptability and introduces subjectivity. To strengthen robustness, future work will

integrate dynamic weight adjustment through machine learning and optimization methods. Approaches such as supervised learning or reinforcement learning could optimize these weights based on empirical threat detection performance, while sensitivity analysis will be used to evaluate how variations in these weights influence the Threat Risk score.

Following the calculation, the Threat Risk score directly informs the entity's risk categorization, defining specific thresholds that translate numerical scores into actionable risk levels. Algorithm 17 illustrates the categorization procedure clearly:

Algorithm 17: Risk Level Determination

Input: Entity identifier EntityID, threat score ThreatRisk from anomaly detection

Output: Updated risk level for EntityID

```
// Determine the risk level based on threshold ranges
Function DetermineRiskLevel(EntityID, ThreatRisk):
    if ThreatRisk ≥ 0.8 then
        UpdateRiskLevel(EntityID, "Critical Risk");
    else if ThreatRisk ≥ 0.6 then
        UpdateRiskLevel(EntityID, "High Risk");
    else if ThreatRisk ≥ 0.4 then
        UpdateRiskLevel(EntityID, "Low Risk");
    else
        UpdateRiskLevel(EntityID, "Normal Risk");
    return CurrentRiskLevel(EntityID);
```

Risk levels are continuously updated based on real-time data inputs, ensuring timely responsiveness to threat dynamics:

- **Critical Risk** (≥ 0.8): Triggers immediate restrictive actions due to high threat confidence.
- **High Risk** ($0.6 \leq$ **score** < 0.8): Activates heightened monitoring and stricter security controls.
- **Low Risk** ($0.4 \leq$ **score** < 0.6): Results in increased vigilance while preserving operational flexibility.
- **Normal Risk** (< 0.4): Indicates standard operating conditions without additional restrictions.

The defined thresholds are based on empirical evaluations and industry-standard cybersecurity practices, designed to effectively balance robust security measures and operational performance. Organizations may adapt these thresholds to their specific operational requirements and security policies, allowing customized responsiveness to their particular security landscape.

The PE integrates a FSM to manage dynamic transitions between security states based on event-driven triggers. The FSM operates by processing events such as minor_anomaly_detected, significant_issue_detected, and anomaly_resolved, which

correspond to real-time security observations. These events dictate transitions between states such as *Normal*, *Alert*, *High Risk*, and *Quarantined*, allowing the system to respond adaptively to changing conditions.

The FSM is defined as a tuple (S, Σ, δ, s_0):

- S: A finite set of states: *Normal*, *Alert*, *High Risk*, *Quarantined*, *Compromised*, *Recovery*.
- Σ: A finite set of events representing observed anomalies or resolutions.
- $\delta : S \times \Sigma \to S$: A state transition function mapping the current state and event to a new state.
- s_0: The initial state, *Normal*.

An event-driven design empowers the PE to respond swiftly and effectively to real-time threats. Although the Threat Risk score influences event generation, the FSM transitions deterministically based on these events, ensuring consistent and predictable state changes.

Figure 7.2 illustrates the Finite State Machine. In particular, when a minor anomaly is detected, the FSM transitions from a *normal* state to an *Alert* state. On the contrary, a critical threat forces the system to go in quarantine, where isolation measures are enforced. The system returns to a normal state when anomalies are resolved.

7.4.6 Finite State Machine (FSM) Event Generation

The FSM transitions between states based on events that are directly tied to the calculated Threat Risk scores. Events such as *minor_ anomaly_ detected*, *significant_ anomaly_ detected*, or *critical_ threat_ detected* are triggered when the Threat Risk score surpasses predefined thresholds, determined as follows:

- **Minor anomaly detected**: Triggered when the Threat Risk score is ≥ 0.4 and < 0.6.
- **Significant anomaly detected**: Triggered when the Threat Risk score is ≥ 0.6 and < 0.8.
- **Critical threat detected**: Triggered when the Threat Risk score is ≥ 0.8.

Initial thresholds (0.4, 0.6, and 0.8) were established manually based on practical judgment, influenced by industry-recognized best practices for anomaly severity classification and security response escalation. Specifically, guidance from frameworks such as the *NIST SP 800* series [**nist-sp800-30r1**] and the *OWASP Risk Rating Methodology* provided conceptual benchmarks [**owasp-risk-rating**] for defining progressive threat levels from minor anomalies to critical threats requiring isolation.

While these thresholds were not derived from formal optimization or machine learning, they were chosen to reflect common principles in risk management such as prioritizing availability in safety-critical IIoT environments and avoiding oversensitization. Within our experimental setup, we qualitatively validated that these

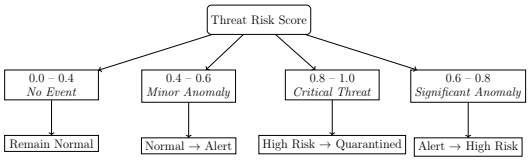


Figure 7.3: Threat Risk Score Mapping to FSM Events

thresholds produced FSM transitions aligned with expected security responses under simulated benign and malicious behaviors.

We acknowledge that this manual, rule-based calibration is a simplification. Future work will explore data-driven threshold refinement using operational telemetry and feedback loops [**wu2024physics**] to improve precision and reduce false positives in dynamic conditions.

As illustrated in Figure 7.3, each Threat Risk interval refers to a distinct event, which drives FSM state transitions.

7.4.7 Policy Creation and Management

The Policy Administration (PA) facilitates the creation, modification, and storage of security policies for various devices and entities within the network. Each policy includes conditions that reference the state of the entity as defined by the PE (*Normal*, *Alert*, *High Risk*, *Quarantined*, *Compromised*, and *Recovery*).

For example, a policy rule may specify:

- **Permit** access to resource R if the entity state is *Normal* and the authentication level is sufficient.
- **Deny** access to sensitive resource S if the entity state is *High Risk* or higher.
- **Require** additional verification steps if the entity state is *Alert*.

These policies are stored in a centralized repository managed by the PA and are retrieved by the PE during the access control decision process.

7.4.8 Meta-Policy Enforcement and Validation

A distinctive feature of the Policy Administration (PA) is its enforcement of meta-policies, which are high-level rules governing the formulation and validation of new or updated policies. These meta-policies ensure that all policies adhere to organizational standards. The validation process considers both semantic correctness and risk alignment.

Policy validation involves the following steps:

1. **Semantic Analysis**: The PA verifies that policies align with high-level objectives and do not conflict with existing rules, ensuring their logical consistency. Meta-policies are pre-compiled into a decision tree structure, where each node represents a condition or constraint. The proposed policy is then abstracted into a feature vector and traverses the decision tree for validation. Conflicts are resolved based on predefined priority levels of meta-policies.

2. **Risk Alignment**: Policies are validated against the organization's risk management principles. Attributes such as entity states, segment sensitivity, and access levels are evaluated to ensure alignment. For instance, higher risk states (*High Risk* or *Quarantined*) correspond to stricter access controls, enforced through meta-policies like `no_write_access_high_risk`.

Algorithm 18: Pre-Compiled Meta-Policy Validation

Input: Meta-policy set $\mathcal{M}$, feature vector $\mathbf{x}$, priority map $\mathbf{P}$
Output: Validation result ValidationResult, list of violations Violations

```
// Validation process for a proposed policy
Function ValidatePolicy(M, x, P):
    Violations ← ∅;
    // Compile meta-policies into decision tree
    T ← CompileToDecisionTree(M);
    // Extract features from proposed policy
    x ← ExtractFeatures(p);
    // Traverse decision tree and check conditions
    foreach node n in T do
        if EvaluateCondition(φ_n, x) = False then
            Append n.meta-policy to Violations;
    // Resolve conflicts based on priority
    Violations ← SortByPriority(Violations, P);
    Violations ← ResolveConflicts(Violations);
    // Determine validation result
    if Violations ≠ ∅ then
        ValidationResult ← False;
    else
        ValidationResult ← True;
    return ValidationResult, Violations;
```

Algorithm 18 formalizes the validation process, ensuring both semantic and risk-based compliance. The meta-policy language is formally structured with clearly defined syntax: each meta-policy comprises a name, type (any/all), and conditions. Conditions are expressed as logical predicates involving entity attributes such as state, access level, and segment sensitivity.

In algorithm 18, $\mathcal{M}$ represents the set of meta-policies, where each meta-policy m includes a condition ϕ_m that must evaluate to **True** for compliance. $\mathbf{x}$ is the feature vector extracted from the proposed policy p, containing relevant attributes such as `state` or `access_level`. T denotes the decision tree constructed from $\mathcal{M}$, where each node n represents a condition ϕ_n for validation. $\mathbf{P}$ is the priority map used to resolve conflicts by assigning precedence to meta-policies. Violations is a list accumulating the meta-policies violated by the proposed policy, and ValidationResult indicates whether the policy complies (**True**) or fails (**False**). Logging mechanisms record violations and outcomes for auditing and debugging purposes.

Conflicts from overlapping meta-policy conditions are resolved systematically using a predefined priority map that ranks meta-policies according to organizational security objectives. In cases of conflict, the highest-priority meta-policy is enforced,

ensuring consistent and predictable policy decisions.

7.4.9 Complexity Analysis

Algorithm 18 addresses scalability through pre-compilation of meta-policies into a decision tree, significantly reducing runtime validation complexity. This design ensures efficiency even as the number of meta-policies grows. Additionally, representing policies as compact feature vectors further minimizes computational overhead, ensuring high scalability and responsiveness in large-scale, dynamic environments. In the following, the complexity is discussed in more detail.

1. **Preprocessing Complexity**: Compiling meta-policies into a decision tree involves parsing and restructuring conditions. Let $|\mathcal{M}|$ be the number of meta-policies and a the average number of attributes per meta-policy. The preprocessing complexity is $O(|\mathcal{M}| \cdot a)$

2. **Runtime Validation Complexity**: Traversing the decision tree has complexity proportional to its depth d, which depends logarithmically on the number of meta-policies in a balanced tree, *i.e.*, $O(d)$ where $d \approx \log(|\mathcal{M}|)$

   Extracting the feature vector $\mathbf{x}$ from the policy has complexity $O(k)$ where k is the number of relevant attributes in the policy

   Thus, the total runtime complexity is $O(d + k) \approx O(\log(|\mathcal{M}|) + k)$

3. **Conflict Resolution Complexity**: Sorting violations by priority involves a complexity of $O(v \cdot \log(v))$ where v is the number of violations. In the worst case, $v = |\mathcal{M}|$, leading to a complexity of $O(|\mathcal{M}| \cdot \log(|\mathcal{M}|))$

The total complexity is finally to $O(|\mathcal{M}| \cdot a + \log(|\mathcal{M}|) + k + |\mathcal{M}| \cdot \log(|\mathcal{M}|))$.

7.4.10 Example

Let us consider a meta-policy `no_write_access_high_risk`, which enforces that entities in the *High Risk* state cannot have write access:

```
{
  "name": "no_write_access_high_risk",
  "type": "any",
  "condition": "state != 'High Risk'
  or access_level != 'full'"
}
```

In this policy, an entity with a `High Risk` must not have the `full access`, preventing it from having a write access.

Let us now consider a proposed policy for `iot_device1` that specificies:

```
{
  "entity_id": "iot_device1",
  "state": "High Risk",
  "access_level": "full"
}
```

To validate this specific policy against the meta-policy:

1. The feature vector is extracted: $\mathbf{x} = \{$state: 'High Risk', access_level: 'full'$\}$.
2. The decision tree evaluates $\phi = (\text{state} \neq' HighRisk') \vee (\text{access\_level} \neq' full')$. This evaluates to **False**.
3. The violation is logged: `"Policy violates meta-policy: no_write_access_high_risk"`.
4. The algorithm returns `ValidationResult = False` and appends the violation to the `Violations` list.

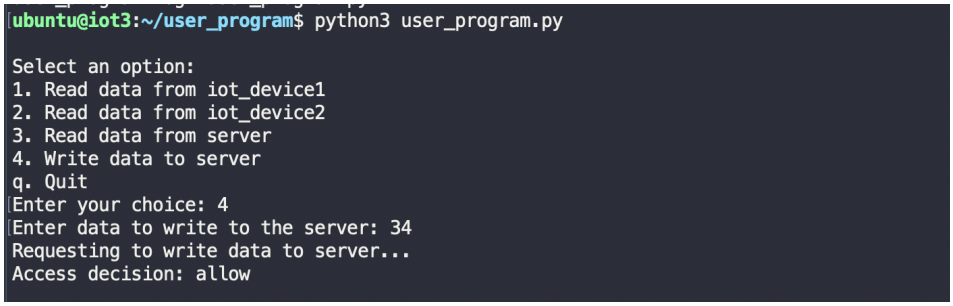
This approach ensures that all proposed policies are rigorously validated against meta-policies, aligning with the principles of Zero Trust Architecture while optimizing resource efficiency.

7.5 Proof of Concept Implementation: a Qualitative Evaluation

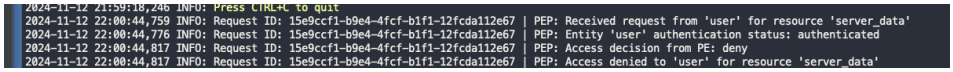
To validate our ZTA framework, we implemented a proof-of-concept (PoC) network using a cluster of virtual machines (VMs) that simulate a controlled network environment. This setup executes key components of the ZTA system, including policy generation, enforcement, and risk assessment, in a small-scale network to analyze access scenarios and policy effectiveness.

Table 7.2: Overview of VM Setup for ZTA Experiment.

Number	Primary Function	Role in Architecture	Specifications
VM1	Policy Administration (PA) and Policy Generator	Manages and updates security policies dynamically	4 CPU, 6 GB RAM, 12 GB disk
VM2	State Checking (PE)	Enforces policies based on device state	4 CPU, 6 GB RAM, 12 GB disk
VM3	Authenticator and Risk Assessment	Evaluates risk based on anomaly detection data	4 CPU, 4 GB RAM, 12 GB disk
VM4	Policy Enforcement Points (PEPs)	Executes updated security policies	4 CPU, 4 GB RAM, 12 GB disk
VM5	Server	Manages authentication status	1 CPU, 2 GB RAM, 12 GB disk
VM6	Simulated User Device	Simulates user access behavior	1 CPU, 2 GB RAM, 12 GB disk
VM7	Simulated IoT Device	Simulates IoT data transmission	1 CPU, 2 GB RAM, 12 GB disk
VM8	Simulated IoT Device	Simulates additional IoT data transmission	1 CPU, 2 GB RAM, 12 GB disk



(a) No ZTA, Allows attacker access if credentials are stolen.



(b) Uses IP and context-aware rules to block unauthorized access attempts.

Figure 7.4: Comparison of Credential Theft Scenarios

Table 7.2 details the VMs used in this setup, outlining their primary functions, roles within the architecture, and specifications.

This PoC setup allows testing interactions between the various ZTA components, supporting simulated access and policy enforcement scenarios in a virtualized environment [243]. We qualitatively evaluate the benefits of the ZTA framework by analyzing the impact of possible attacks. We consider the following scenario:

1. The user has password-protected access to the server.
2. The user has read-only access to Device 1 but no access to Device 2.
3. Each IoT device can send data to the server at defined intervals.

7.5.1 Credential Theft and Unauthorized Server Access

Credential theft is a prevalent and critical attack vector where attackers exploit stolen user credentials to gain unauthorized access to sensitive systems. This scenario mimics real-world incidents like phishing attacks or brute-force credential compromises, which often bypass traditional password-based security systems. The lack of contextual verification mechanisms exacerbates this vulnerability, allowing attackers to leverage valid credentials undetected. Such breaches can lead to severe consequences, including unauthorized data exfiltration, system compromise, or further lateral movement within the network.

Without ZTA: The attacker can authenticate and gain full access to the server (Figure 7.4a).

With ZTA: ZTA policies restrict access based on IP and user context, blocking unauthorized attempts from unfamiliar locations (Figure 7.4b).

7.5.2 Insider Threat and Unauthorized Device Access

Insider threats pose significant risks to organizational security by exploiting legitimate access to systems. In this scenario, a user with authorized access to Device

```
Select an option:
1. Read data from iot_device1
2. Read data from iot_device2
3. Read data from server
4. Write data to server
q. Quit
Enter your choice: 2
Requesting data from 'iot_device2'...
Access decision: allow
```

(a) Without ZTA, allowing unauthorized access to sensitive devices

```
INFO: Request ID: 8eeaf93a-e395-41ae-86d2-7cfcff90c071 | PEP: Received request from 'user' for resource 'iot_device2_data'
INFO: Request ID: 8eeaf93a-e395-41ae-86d2-7cfcff90c071 | PEP: Entity 'user' authentication status: authenticated
INFO: Request ID: 8eeaf93a-e395-41ae-86d2-7cfcff90c071 | PEP: Access decision from PE: deny
INFO: Request ID: 8eeaf93a-e395-41ae-86d2-7cfcff90c071 | PEP: Access denied to 'user' for resource 'iot_device2_data'
```

(b) Applies role- and context-based policies to limit access and block suspicious behavior.

Figure 7.5: Comparison of Insider Threat Scenarios

```
2024-11-12 22:08:30,906 INFO: Request ID: d3572511-ff30-4c40-ba31-d31e36d97ec0 | PEP: Received request from 'iot_device1' for resource 'send_data_to_server'
2024-11-12 22:08:30,923 INFO: Request ID: d3572511-ff30-4c40-ba31-d31e36d97ec0 | PEP: Entity 'iot_device1' authentication status: authenticated
2024-11-12 22:08:30,947 INFO: Request ID: d3572511-ff30-4c40-ba31-d31e36d97ec0 | PEP: Access decision from PE: deny
2024-11-12 22:08:30,947 INFO: Request ID: d3572511-ff30-4c40-ba31-d31e36d97ec0 | PEP: Access denied to 'iot_device1' for resource 'send_data_to_server'
2024-11-12 22:08:30,948 INFO: 192.52.32.20 - - [12/Nov/2024 22:08:30] "POST /handle_request HTTP/1.1" 200 -
2024-11-12 22:08:36,959 INFO: Request ID: e624838b-1dfa-4a66-929e-e87b549ba919 | PEP: Received request from 'iot_device1' for resource 'send_data_to_server'
2024-11-12 22:08:36,965 INFO: Request ID: e624838b-1dfa-4a66-929e-e87b549ba919 | PEP: Entity 'iot_device1' authentication status: authenticated
2024-11-12 22:08:36,987 INFO: Request ID: e624838b-1dfa-4a66-929e-e87b549ba919 | PEP: Access decision from PE: deny
2024-11-12 22:08:36,987 INFO: Request ID: e624838b-1dfa-4a66-929e-e87b549ba919 | PEP: Access denied to 'iot_device1' for resource 'send_data_to_server'
2024-11-12 22:08:36,988 INFO: 192.52.32.20 - - [12/Nov/2024 22:08:36] "POST /handle_request HTTP/1.1" 200 -
2024-11-12 22:08:42,995 INFO: Request ID: 0c6a85d1-fb66-4cea-b848-76a5b92a717a | PEP: Received request from 'iot_device1' for resource 'send_data_to_server'
2024-11-12 22:08:43,009 INFO: Request ID: 0c6a85d1-fb66-4cea-b848-76a5b92a717a | PEP: Entity 'iot_device1' authentication status: authenticated
2024-11-12 22:08:43,038 INFO: Request ID: 0c6a85d1-fb66-4cea-b848-76a5b92a717a | PEP: Access decision from PE: deny
2024-11-12 22:08:43,038 INFO: Request ID: 0c6a85d1-fb66-4cea-b848-76a5b92a717a | PEP: Access denied to 'iot_device1' for resource 'send_data_to_server'
```

Figure 7.6: ZTA Reaction to DoS Attack Scenario

1 attempts to gain unauthorized access to Device 2, bypassing conventional network controls. This scenario mimics real-world incidents where insiders abuse their permissions or credentials to access sensitive resources, a critical challenge for traditional security architectures. The lack of fine-grained, context-aware access control exacerbates the risk of exposing sensitive data or critical infrastructure to unauthorized users.

Without ZTA: Basic network security is bypassed, allowing access to Device 2 (Figure 7.5a). *With ZTA*: Role-based policies deny access to Device 2, and the Policy Engine flags any anomalous access attempts (Figure 7.5b).

7.5.3 Compromised IoT Device and DoS Attack

In this scenario, an IoT device is compromised by an attacker and begins sending excessive amounts of data to the server, resulting in a Denial-of-Service (DoS) attack. This type of attack mimics real-world incidents where IoT devices, often lacking robust security controls, are exploited to overwhelm critical infrastructure. Such attacks are critical to address as they can cause server downtime, disrupt operations, and impact the availability of services.

Without ZTA: The server becomes overwhelmed, leading to potential downtime.

With ZTA: Rate-limiting policies prevent excessive data from any IoT device, mitigating the DoS attack.

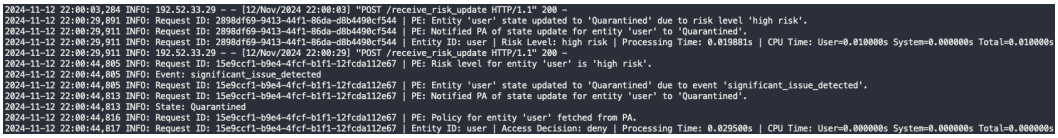


Figure 7.7: ZTA Reaction to Suspicious User Behavior

7.5.4 Suspicious User Behavior and Anomaly Detection

This scenario involves a compromised user account engaging in actions that deviate significantly from its standard behavior patterns . Such behavior may include accessing sensitive resources, executing commands outside the user's typical scope, or initiating anomalous transactions. This scenario mimics real-world incidents where compromised credentials or insider threats exploit legitimate access to cause harm. Detecting these deviations is critical to preventing unauthorized activities and mitigating potential damage.

Without ZTA: Abnormal actions remain undetected, potentially causing damage.

With ZTA: Context-aware anomaly detection flags deviations, prompting the Policy Engine to require multi-factor authentication or deny further access.

7.6 Quantitative Evaluation of Proposed ZTA

The PoC environment, detailed in Section 7.5, was configured to simulate typical access scenarios within an IIoT context under normal network operations. The implementation included critical ZTA components such as the PA, PE, and PEP across a cluster of virtual machines.

It is important to clarify that our ZTA framework does not enforce policies at the granularity of individual packets traversing the network. Instead, policy enforcement occurs primarily at session initiation and at strategic checkpoints or upon detecting anomalous behavior. Specifically, a lightweight probe passively inspects network traffic without directly interfering with routing, collecting contextual information and real-time threat indicators. Based on this information, the Policy Enforcement Point (PEP) enforces security policies proactively at connection setup and reactively when anomalies are detected. This approach minimizes computational overhead and maintains high security standards without unnecessarily impacting network performance.

The evaluation aimed to measure performance metrics, including latency, CPU usage, and memory utilization, during normal network operations. These metrics provide insights into the baseline performance of the ZTA framework, demonstrating its efficiency and scalability in regular conditions.

7.6.1 Latency

Figure 7.8 shows the comparison of latency with and without ZTA across different request rates. Latency was measured as the average time required to process access requests under varying network loads. During normal operations, latency remained within acceptable bounds. Without ZTA, latency was consistently low and averaged approximately 45 ms across a range of request rates. With ZTA, latency increased

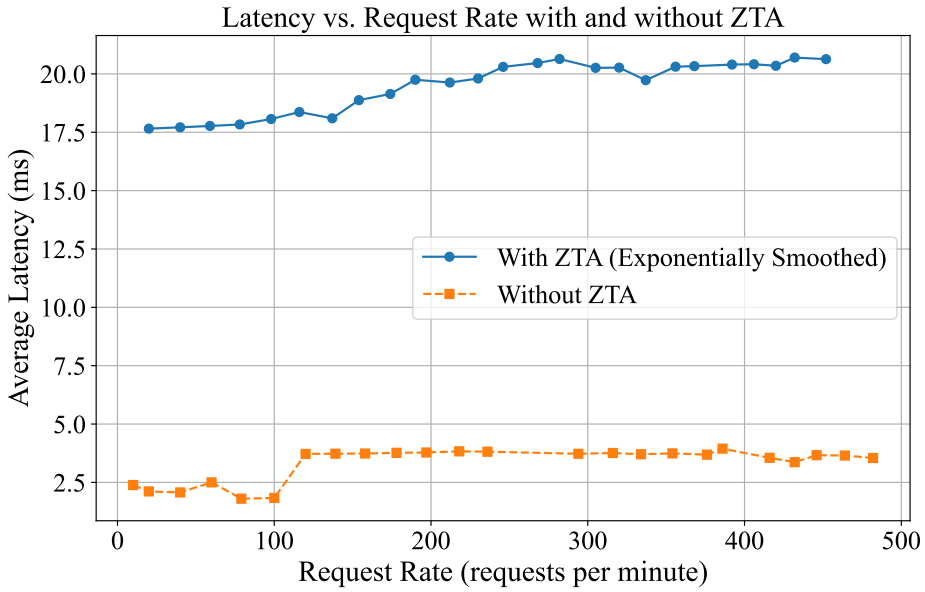


Figure 7.8: Latency vs. Request Rate with and without ZTA

due to context validation and policy enforcement checks but remained well below the threshold for near-real-time IIoT operation. The average latency under ZTA peaked at approximately 141 ms at the highest simulated traffic levels. This controlled rise in latency demonstrates the effectiveness of the ZTA framework in maintaining operational responsiveness under increasing workload.

7.6.2 CPU and Memory Utilization

Figure 7.9 illustrates CPU consumption relative to the request rate, demonstrating consistent and predictable scaling as traffic volume increases. To assess the computational efficiency of the ZTA framework under normal operation, CPU usage was recorded across all major components of the PoC environment.

In deployments without ZTA, CPU utilization ranged from 1.5% to 7.0%, reflecting minimal computational burden associated with basic access control mechanisms. In contrast, the ZTA deployment began at approximately 16% and scaled linearly with traffic, reaching a maximum of 26.4% at higher request rates. This increase is primarily attributable to the additional workload introduced by real-time risk assessment, policy lookup, and state-checking logic executed within the PEP and PE.

To further clarify the distribution of load across system components, we conducted per-VM profiling under typical traffic conditions. The observed 26.4% peak represents a balanced workload across all CPU cores, avoiding saturation on any single thread. Average CPU utilization per virtual machine was as follows:

- VM2 (PE and State Checking): $25.1\% \pm 3.8\%$
- VM3 (Authenticator and Risk Assessment): $21.3\% \pm 4.0\%$

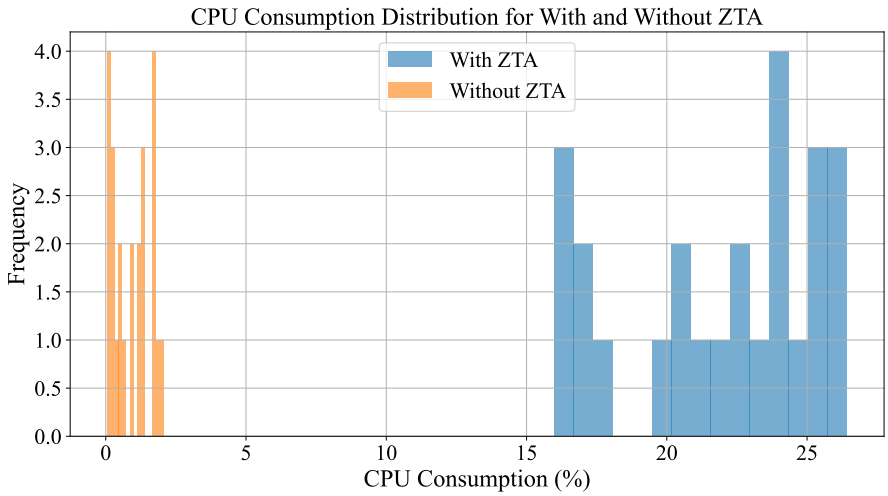


Figure 7.9: CPU Scaling With/Without ZTA

- VM4 (Policy Enforcement Points - PEPs): $26.4\% \pm 5.1\%$
- VM5 (Server): $12.2\% \pm 2.3\%$
- VM6 (Simulated User Device): $8.5\% \pm 1.4\%$
- VM7 and VM8 (Simulated IoT Devices): $6.8\% \pm 1.1\%$

The VMs simulating IoT endpoints (VM7 and VM8) executed lightweight scripts that periodically generated and transmitted synthetic telemetry to the server, simulating typical IIoT sensor communication patterns. This emulation enabled us to benchmark system behavior under realistic edge-to-core traffic flows without introducing unnecessary complexity at this stage. In future work, we intend to incorporate more representative IIoT simulators and heterogeneous device types to further refine performance profiling across protocol stacks.

Memory usage remained stable throughout all test runs. Core ZTA components particularly the PE and PEP used modest memory overhead to retain per-session context and temporary policy caches. Overall memory consumption remained within the 22–24% range across core components, indicating effective resource allocation. These results confirm that the proposed framework provides scalable and context-aware policy enforcement without imposing excessive resource demands, making it suitable for deployment on mid-tier edge gateways and IIoT control nodes.

7.6.3 Performance Metrics and Threat Response

To comprehensively evaluate the operational overhead and dynamic threat response capabilities of the proposed ZTA, we conducted a detailed performance analysis across three distinct scenarios: (1) **Normal Operation**, representing standard conditions; (2) **High-Frequency Access**, simulating elevated request rates; and (3) a targeted **Denial-of-Service (DoS) Attack**, simulating adversarial behavior. We

measured key performance metrics (CPU usage, memory consumption, latency, network throughput, and packet loss) across critical system components: User, Server, PEP, PE.

Table 7.3 summarizes the average performance metrics, while Figures 7.10-7.12 visually depict the system's behavior under each scenario.

Table 7.3: Performance Metrics Across Scenarios.

Metric	Entity	Normal	HighFreq	DoS
CPU Usage (%)	User	14.2	22.3	27.5
	Server	11.8	18.5	19.7
	PEP	13.1	20.1	22.4
	PE	12.7	19.2	21.1
Memory Usage (%)	User	23.5	28.6	31.2
	Server	21.1	24.3	26.4
	PEP	22.8	26.5	28.1
	PE	21.7	25.8	27.9
Throughput (Mbps)	User	42.8	39.2	35.8
	Server	42.8	39.2	35.8
	PEP	41.7	37.5	34.6
Latency (ms)	User	120.5	160.7	210.4
	PEP	130.2	175.9	248.9
Packet Loss (%)	User	2.1	8.5	16.5
	PEP	3.4	9.8	18.3

Under Normal Operation, the ZTA framework exhibited stable and efficient resource management. As shown in Figure 7.10, CPU usage remained modest across components, averaging below 15%. Memory consumption followed a similar trend, as illustrated in Figure 7.11, remaining around 22% on average. Network latency, was low 120.5 ms for the User and 130.2 ms for the PEP while Figure 7.12 shows throughput was high (42.8 Mbps for User and Server), with minimal packet loss (2.1% for User, 3.4% for PEP). These results confirm that the system imposes minimal overhead during standard IIoT operations.

In the High-Frequency Access scenario, the system experienced moderate stress due to increased request rates. CPU usage increased noticeably (see Figure 7.10), reaching 22.3% for the User and approximately 20% for the PEP. Figure 7.11 shows memory usage rose to nearly 28.6% (User) and 26.5% (PEP). Latency increased by about one-third, with Table 7.3 over latency showing a jump to 160.7 ms (User) and 175.9 ms (PEP). As seen in Figure 7.12, throughput slightly declined (8%), and packet loss increased to 8.5% and 9.8% respectively. Despite these increases, system performance remained within operational limits.

During the simulated DoS Attack, the attacker generated traffic exceeding 30 requests per 10 seconds, overwhelming normal thresholds. The PEP's anomaly detection module promptly identified this pattern as a data flooding attack with 95% confidence. The resulting alert triggered a state transition in the FSM to High Risk, activating strict access controls via the Risk Assessment module.

The attack's impact on performance was substantial: as shown in Figure 7.10, CPU usage surged to 27.5% (User) and 22.4% (PEP). Memory usage also spiked

(Figure 7.11), reaching 31.2% (User) and 28.1% (PEP). Latency, illustrated in Table 7.3 over latency, escalated sharply to 210.4 ms (User) and 248.9 ms (PEP). Throughput, as shown in Figure 7.12, dropped to 35.8 Mbps (User and Server), and packet loss rose to 16.5% (User) and 18.3% (PEP). Despite this degradation, the Server and Policy Engine sustained stable performance, indicating effective containment of the compromised entity.

These comprehensive results confirm the ZTA framework's capacity for real-time threat detection, dynamic policy enforcement, and resilience under both elevated workload and active adversarial conditions. The architecture isolates threats without compromising the stability of the broader system.

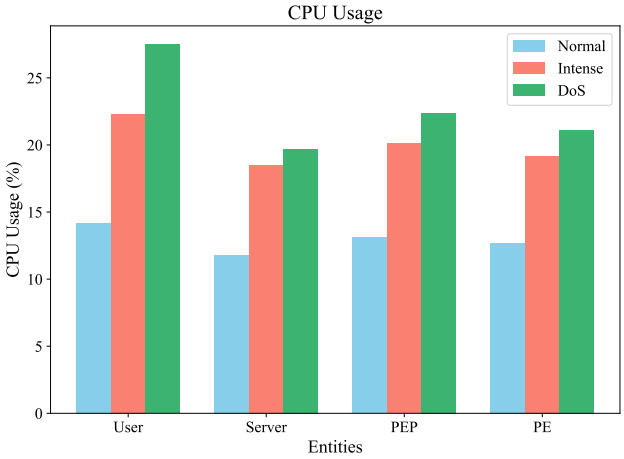


Figure 7.10: CPU Usage across scenarios

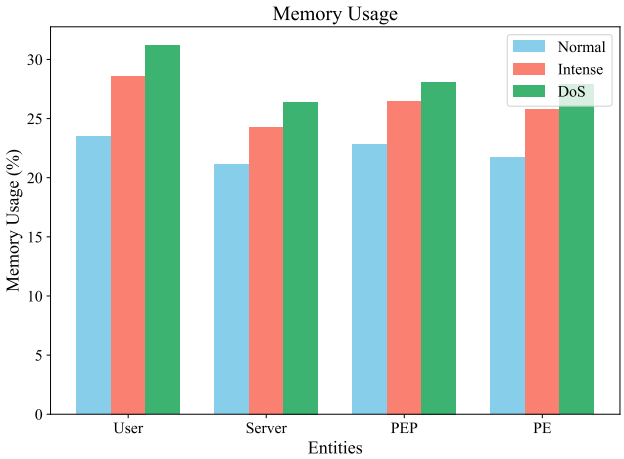


Figure 7.11: Memory Usage across scenarios

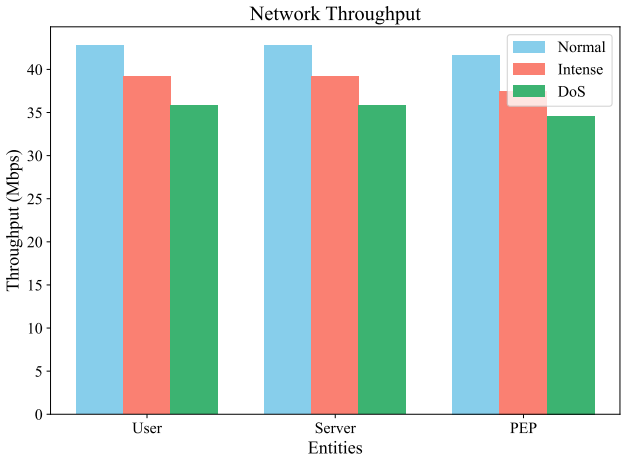


Figure 7.12: Network Throughput across scenarios

7.6.4 Scalability, Interoperability, and Edge Deployment Considerations

To further enhance the practical deployment of the proposed ZTA framework in IIoT environments, several key considerations have been identified for future work.

Edge Device Deployment Resource constraints, such as limited CPU, memory, and energy availability, are critical challenges for IIoT edge devices. While our proof-of-concept focused on server-grade virtual machines, future work will involve implementing lightweight PEPs and context probes on resource-constrained devices such as Raspberry Pi and microcontrollers. Optimization strategies, including offloading intensive computations to nearby edge servers and minimizing local policy enforcement overhead, will be investigated to ensure scalability to highly distributed, resource-limited environments.

Multi-Tenancy Support In industrial ecosystems, multi-tenancy is often required where multiple independent organizations or operational domains coexist. Scaling the ZTA framework to support multiple tenants will involve isolated policy domains, federated risk assessments, and tenant-aware PEPs/PEs to ensure secure, logical separation while sharing underlying infrastructure.

Protocol Interoperability Although our current implementation operates over standard TCP/IP, real-world IIoT systems rely heavily on specific industrial protocols such as MQTT and OPC UA. Future iterations of the framework will integrate protocol adapters and semantic translators to support these standards natively, allowing for dynamic, context-aware security policies across heterogeneous communication stacks.

Scale-Out of PEPs and PEs Supporting large-scale IIoT deployments requires scalable PEP and PE architectures. Future designs will employ hierarchical or distributed control models to allow PEPs and PEs to scale horizontally, with mecha-

nisms for distributed policy synchronization, context aggregation, and decentralized anomaly detection to maintain performance and resilience across large, dynamic networks.

7.7 Conclusion

This chapter introduced an advanced Dynamic ZTA, specifically designed to enhance security in dynamic and context-rich IIoT environments. The proposed dynamic ZTA framework significantly advances traditional static approaches by integrating real-time context-awareness and continuous threat assessments into access control mechanisms. By dynamically adapting security policies in real-time, the presented ZTA effectively addresses critical vulnerabilities and limitations associated with rigid, static policy enforcement models, thus ensuring robust protection against evolving threats.

The primary advantages of the proposed dynamic ZTA lie in its adaptability and responsiveness, allowing IIoT systems to respond swiftly to emerging threats and changes in operational context. It substantially enhances operational security by integrating continuous monitoring and immediate policy adjustments, which are crucial for complex industrial environments characterized by dynamic interactions among multiple devices and users.

However, certain limitations persist, primarily related to the complexity and resource demands of real-time data analysis and context assessment. Implementing this dynamic architecture in highly resource-constrained environments may present operational challenges, necessitating further optimization and targeted adaptation strategies.

In conclusion, this chapter contributes significantly to developing a robust and adaptable security framework that addresses key gaps in traditional access control methodologies within IIoT networks. The dynamic ZTA model sets the foundation for further enhancements presented in the next chapter, where Chapter 8 will explore distributed, blockchain-based mechanisms for secure policy negotiation. This subsequent approach aims to complement and extend the capabilities introduced here by adding further layers of decentralized, verifiable, and transparent security management to the overall security ecosystem.

Chapter 8

Distributed Zero Trust Architecture

Contents

Publications:

- Fatemeh Stodt, Philipp Ruf, Christoph Reich, and Fabrice Théoleyre (2025). "Distributed Zero Trust Architecture Based on Policy Negotiation Secured by DPP in Blockchain". In: *Annals of Telecommunications*. Under review.

8.1 Introduction

As IIoT ecosystems evolve into distributed and multi-stakeholder environments, static or centrally managed trust frameworks struggle to meet the growing demands

for scalability, autonomy, and interoperability. Secure collaboration across independently administered industrial domains requires dynamic, verifiable trust mechanisms that can adapt to contextual policies, varying operational goals, and regulatory constraints.

Traditional ZTA, even when extended with dynamic access control mechanisms, often assume centralized control over policies and identities. In decentralized industrial settings, such assumptions break down. Trust must be established and managed in a distributed, auditable, and transparent manner while also ensuring security, privacy, and operational flexibility.

To address these challenges, this chapter introduces a novel contribution that extends Zero Trust principles into a fully decentralized and blockchain-enhanced trust negotiation framework:

This contribution builds upon and integrates the lightweight blockchain architecture and the dynamic Zero Trust model introduced in earlier chapters. It presents a unified framework that enables secure, automated policy negotiation among distributed industrial entities. Our approach uniquely combines the immutability and auditability of blockchain with the real-time policy adaptation capabilities of dynamic ZTAs.

Previous decentralized trust models exhibit various limitations. For instance, DistriTrust [244] decentralizes policy decisions using threshold cryptography but faces scalability issues and synchronization challenges across distributed Policy Decision Points (PDPs). Similarly, Xie *et al.*'s distributed ZTA [245] integrates federated learning and blockchain, enhancing resilience but introducing complexity and scalability concerns for large industrial networks. Mahalle *et al.* [56] propose a capability-based model tailored for IoT access control but lack decentralized decision-making.

Table 8.1 summarizes key differentiators between these methods and our proposed framework, highlighting advancements in scalability, flexibility, privacy, and transparency.

Table 8.1: Comparison of Key Capabilities in Distributed Policy Architectures

Capability	**[244]**	**[245]**	**[56]**	**Our Approach**
Dynamic Policy Negotiation	✗	✗	✗	✓
Privacy-Preserving Access Control	✗	✓	✓	✓
Decentralized Policy Decision	✓	✓	✗	✓
Adaptive Regulatory Compliance	✗	✗	✓	✓
Attribute-Based Privacy Controls	✗	✗	✓	✓
Fine-Grained Trust Evaluation	✗	✓	✗	✓
Blockchain Consensus	✓	✓	✗	✓
Multi-party Negotiation	✗	✗	✗	✓
Quantitative Risk Assessment	✗	✗	✗	✓
Smart Contract Anchoring	✗	✓	✗	✓

In our framework, capabilities are utilized for system authorization, ensuring requester and requestee permissions alignment.

At the core of this integration is the Digital Product Passports (DPP) mechanism, which ensures that negotiated policies remain verifiable, context-aware, and privacy-preserving. This synergy allows for transparent and adaptive trust negotiation, even across independently governed industrial domains.

By leveraging the strengths of both the lightweight blockchain infrastructure and the dynamic ZTA introduced earlier, the proposed Distributed ZTA delivers a robust, scalable, and context-sensitive trust architecture. It provides a foundation for secure, cross-domain collaboration addressing one of the most critical challenges in the future of industrial security.

Addresses Research Questions:

- **What are the challenges in integrating blockchain and Zero Trust principles into a cohesive security framework for IIoT?**
- **How can identity management, blockchain, and anomaly detection be integrated into a cohesive Distributed ZTA framework for securing IIoT networks?**

8.2 Proposed Distributed ZTA Framework

We propose to enable secure and privacy-preserving collaboration in dynamic and cross-company environments (Figure 8.1). In such a situation, organizations periodically or spontaneously require data or service sharing while preserving their internal policies and maintaining robust security. To achieve this, our framework integrates the ZTA principles with a blockchain-enabled decentralized policy management system.

Each participating organization operates its own ZTA system, with a PE and PEPs managing local rules and access control. The hierarchical blockchain architecture mirrors the roles of ZTA components as follows:

1. **Full Nodes (PEs):** These nodes act as Policy Engines, responsible for policy evaluation, decision-making, and storing immutable consensus data on the blockchain. Full Nodes ensure decentralized management of policies across organizations.

2. **Middle Nodes (PEPs):** These nodes function as Policy Enforcement Points, enforcing access decisions and validating policy adherence. They aggregate and process updates from Light Nodes and interact with Full Nodes for consensus and enforcement.

3. **Light Nodes (Entities):** These nodes represent devices, users, or systems that interact with the blockchain. They store DPPs and Data Access Passports (DAPs) locally while submitting policy-related information to Middle Nodes for processing.

DPPs encapsulate key metadata about devices, products, and entities in the network such as configuration sets, maintenance schedules, and lifecycle data. They also include a policy layer, the DAP, which defines access rules and constraints. These passports preserve local policies and enable secure collaboration in distributed environments. Additionally, policies regarding data access are stored in a dedicated layer referred to as the DAP. These passports ensure local policies remain intact and secure, enabling safe collaboration in dynamic and distributed environments.

The DPPs are generated centrally within each organization's network and pushed to a blockchain, utilizing a privacy-preserving mechanism such as a viewing key. Only authorized members can access the content of the DPP and DAP, ensuring that sensitive policy information is secure. This design safeguards internal rules while providing an immutable proof of negotiation and consensus for cross-company collaboration.

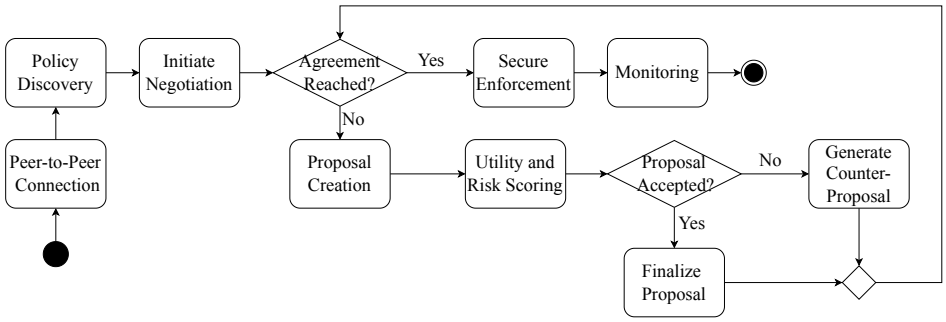


Figure 8.1: Proposed Method Data Flow.

8.2.1 Security Properties Provided by the Blockchain Architecture

The proposed blockchain-enhanced architecture provides the following core security guarantees:

- **Integrity and Authenticity of Policies:** Every policy proposal or update is signed by authorized PEs and committed to an immutable blockchain ledger. This prevents tampering and allows for verification of origin.
- **Confidentiality of Policy Data:** Only metadata or encrypted policy references are stored on-chain. Sensitive policy contents remain off-chain, encrypted using viewing keys and accessible solely to authorized peers.
- **Non-Repudiation and Accountability:** All proposals and consensus agreements are signed and recorded on-chain. No participant can deny past actions or falsely claim policy states.
- **Availability of Policy State:** Replication across nodes ensures that policy history and current agreements remain accessible, even in cases of node failure or network partitioning.

These security properties are essential to uphold the Zero Trust principles of verifiability, transparency, and decentralization across organizations.

8.2.2 Decision-Making Functions

Quantifiable Objectives

To systematically assess and negotiate policy proposals across organizational boundaries, we define a set of quantifiable objectives. These objectives reflect the core

requirements for secure, compliant, and trustworthy inter-organizational cooperation, and are formulated in alignment with established best practices in security, governance, legal compliance, and operational integrity.

Each objective is associated with a normalized metric in the $[0, 1]$ range, facilitating integration into a multi-objective utility function. This structure supports transparent policy comparison and optimization, while enabling risk-aware compliance evaluation and adaptive trust negotiation.

1. **Security Compliance Objective (f_{security}):**

   Measures adherence to baseline security practices derived from internationally accepted standards such as ISO/IEC 27001 [246]. This objective evaluates the extent to which a partner's policy satisfies required technical and organizational safeguards, such as access control, incident response, and encryption.

$$f_{\text{security}}(x) = \frac{\text{Number of implemented baseline security controls}}{\text{Total baseline controls}}$$

2. **Legal and Regulatory Compliance Objective ($f_{\text{regulatory}}$):**

   Assesses the alignment of policy proposals with applicable legal and regulatory frameworks, such as GDPR for European data protection or CCPA for Californian regulations. Compliance with these regulations is essential to ensure lawful processing, user rights assurance, and organizational accountability [247].

$$f_{\text{regulatory}}(x) = \frac{\text{Number of regulatory requirements met}}{\text{Total applicable requirements}}$$

3. **Corporate Governance Alignment Objective ($f_{\text{governance}}$):**

   Evaluates conformity with internal governance directives including intellectual property rules, organizational risk policies, and strategic control mechanisms. Governance alignment is a key enabler of harmonized policy negotiation and accountability [248].

$$f_{\text{governance}}(x) = \frac{\text{Number of internal governance criteria met}}{\text{Total governance criteria}}$$

4. **Data Privacy and Sharing Control Objective (f_{privacy}):**

   Captures how effectively data handling practices align with privacy-preserving principles, including encryption, data minimization, and access control. It also reflects the granularity of control offered for sharing sensitive information across organizations [247].

$$f_{\text{privacy}}(x) = \frac{\text{Number of protected sensitive fields}}{\text{Total sensitive fields requested}}$$

5. **Cultural and Trust-Related Objective (f_{trust}):**

   Quantifies interpersonal trust factors grounded in prior collaborations, transparency practices, auditability, and partner reputation. Trust models are critical in federated or decentralized systems to reduce negotiation friction and uncertainty [249].

$$f_{\text{trust}}(x) = \frac{\text{Trust points scored from prior collaborations and audits}}{\text{Maximum trust score achievable}}$$

6. **Operational Efficiency Objective ($f_{\text{efficiency}}$):**

   Represents adherence to performance guarantees such as uptime, response time, or throughput, often dictated by service level agreements (SLAs). High operational efficiency supports reliable execution of shared policies and dynamic workloads [250].

$$f_{\text{efficiency}}(x) = \frac{\text{Achieved performance metric (e.g., uptime or throughput)}}{\text{Target performance metric}}$$

Utility Function

To aggregate diverse policy evaluation criteria into a coherent decision-making framework, we adopt a weighted sum utility function. This approach offers several key advantages: it is interpretable, computationally efficient, and well-suited for multi-objective scenarios with normalized metrics. It enables each PE to prioritize objectives according to organizational preferences while preserving comparability across proposals.

$$U = \sum_{i=1}^{n} w_i \cdot f_i(x_i) \tag{8.1}$$

Where:

- U is the total utility of the proposal.
- w_i is the weight assigned to objective i, reflecting its relative importance.
- $f_i(x_i)$ is the normalized value of objective i, representing how well it is satisfied.
- n is the total number of objectives under consideration.

This additive model is chosen for its balance between simplicity and expressiveness. It allows decision-makers to transparently encode strategic trade-offs, adapt to context-specific risk tolerances, and incorporate human-in-the-loop preferences. Moreover, its linearity makes it compatible with optimization techniques and game-theoretic analysis used in policy negotiation frameworks.

Security Rationale. The use of these decision-making functions (utility, risk, and compliance) guarantees that only policy proposals meeting defined trust, security, and regulatory thresholds are accepted. Any attempt to bypass requirements would result in rejection during evaluation, with tamper-evident records created through blockchain anchoring.

8.2.2.1 Risk Assessment Function

To complement the utility-based evaluation, a dedicated risk assessment function is introduced to identify proposals that, while beneficial in terms of utility, may still pose unacceptable security or compliance threats. Unlike the utility function, which measures how well a proposal aligns with strategic or operational goals, the risk score captures the potential for adverse consequences ensuring that high-utility proposals are not accepted blindly without regard to their associated vulnerabilities.

Risk is quantified as a function of three parameters: the estimated likelihood of a threat materializing, the potential impact of that threat, and the effectiveness of existing mitigation strategies. These parameters are derived from historical incident data, domain-specific threat models, and predefined control baselines. The risk associated with a proposal is calculated as:

$$R = \frac{\text{ThreatLikelihood} \times \text{Impact}}{\text{MitigationFactor}} \tag{8.2}$$

Where:

- R is the computed risk score for the proposal.
- *ThreatLikelihood* denotes the probability of an adverse security event (e.g., data breach or policy violation).
- *Impact* quantifies the severity of such an event in terms of data sensitivity, regulatory consequences, or service disruption.
- *MitigationFactor* reflects the effectiveness of protective controls in place, such as encryption, multi-factor authentication, or network isolation.

This risk score plays a decisive role during the negotiation filtering stage: proposals yielding high utility but with risk scores exceeding a predefined threshold are automatically rejected or flagged for human review. This ensures that collaboration does not come at the expense of security assurance or regulatory compliance.

8.2.2.2 Compliance Function

The compliance of a proposal is evaluated using:

$$C = \frac{\text{SatisfiedRequirements}}{\text{TotalRequirements}} \tag{8.3}$$

Where:

- C is the compliance score of the proposal.
- *SatisfiedRequirements* is the number of compliance requirements met by the proposal.
- *TotalRequirements* is the total number of compliance requirements.

A higher compliance score indicates better alignment with regulatory and organizational policies.

Algorithm 19 implements the negotiation process by orchestrating secure collaboration, dynamic policy evaluation, and iterative proposal refinement. It ensures

Algorithm 19: Cross-Company Policy Negotiation and Update

Input: Trigger for cross-company collaboration request
Output: Updated policies stored securely in local systems and blockchain

Phase 1: Initiation of Collaboration;
PEs of participating organizations initiate secure communication;
`EstablishSecureConnection();` `// e.g., mutual TLS handshake`
Validate identities and ensure communication integrity;
Phase 2: Accessing Blockchain-Stored Policies;
PEs and PEPs connect to the blockchain;
`RetrievePolicies();` `// Obtain DAPs associated with relevant entities`
`VerifyPolicies(); // Verify policy integrity, signatures, and authorization status`
Phase 3: Negotiation Phase;
Initialize utility threshold τ_U, risk threshold τ_R, and timeout;
while ***Consensus not reached and timeout not exceeded*** **do**
    foreach ***proposal received*** **do**
        Compute utility: $U = \sum_{i=1}^{n} w_i \cdot f_i(x_i)$;
        Compute risk: $R = \frac{\text{ThreatLikelihood} \times \text{Impact}}{\text{MitigationFactor}}$;
        if $U \geq \tau_U$ ***and*** $R \leq \tau_R$ **then**
            Accept proposal;
        else
            `GenerateCounterProposal();` `// Adjust` x_i `to improve` U `or reduce` R
if ***Consensus is reached*** **then**
    Proceed to update policies;
else
    Abort negotiation and optionally re-initiate;
    return;
Phase 4: Updating Local Policies and Blockchain;
`UpdateLocalPolicies();` `// Apply final policies to PEs and PEPs`
Update associated DAPs in DPP structure;
`UpdateBlockchain();` `// Record updated DPPs with access controls`
Guarantee immutability and verifiability through blockchain ledger;
End of Process;

that each participating entity aligns proposals with its priorities while maintaining acceptable risk levels and compliance standards. Upon successful convergence, agreed policies are updated both locally and on-chain, preserving integrity and auditability across the federation.

Resilience Rationale. The risk function protects the negotiation from reckless or adversarial proposals. For example, a malicious peer offering overly permissive access without adequate mitigations would produce a high risk score $R > \tau_R$, causing rejection. Risk assessments thus form a safeguard layer.

Parameter	Description
p	number of PEs
k	number of policies per PE
r	negotiation rounds
$\|\text{pol}\|$	policy size in bytes

Table 8.2: Notation

Resource	Component	Asymptotic Complexity (per PE)
Computational (CPU)		
	Phase 1: TLS handshake	$\Theta(p)$
	Phase 2: Policy retrieval and verification	$\Theta(k)$
	Phase 3: Policy negotiation rounds	$\Theta(r \cdot p)$
	Phase 4: BFT consensus commit	$\Theta(p^2)$
Memory (Storage)		
	Working memory	$\Theta(1)$
	Cached policies	$\Theta(k \cdot \|\text{pol}\|)$
	On-chain policy replicas	$\Theta(k \cdot \|\text{pol}\|)$ per node
Network Communication		
	TLS handshakes	$2p(p-1)$ packets (constant size)
	Proposal broadcast	$\Theta(p^2 \cdot \|\text{pol}\|)$
	Final consensus commit	Equivalent to one proposal

Table 8.3: Computational, memory, and network complexity analysis

8.2.3 Computational, Space, and Network Complexity

Table 8.2 reminds our notation. For a typical supply-chain consortium scenario with 10 PEs (p), 100 policies per PE (k), 5 negociation rounds (r), and $\approx 1,\text{kB}$ per policy ($|\text{pol}|$), the computational, memory, and network requirements remain modest:

- **Computational Complexity:** The most demanding phase (Phase 3) involves $r \cdot p = 50$ evaluations. With approximately six scalar operations per evaluation, this remains computationally trivial and executes in under a millisecond on commodity hardware.
- **Memory Requirements:** Cached policy storage per PE totals around $k \cdot |\text{pol}| \approx 100\,\text{kB}$, a negligible footprint manageable even on resource-constrained devices such as a Raspberry Pi.
- **Network Overhead:** Negotiation traffic accumulates to $r \cdot p^2 \cdot |\text{pol}| = 5 \times 100 \times 1\,\text{kB} \approx 0.5\,\text{MB}$, a minimal amount when compared to standard firmware updates.

- **Consensus Commit Overhead:** An additional one-time burst of approximately p^2 data transmissions ($\approx 100\,\text{kB}$), paid only once and not per round, ensuring negligible sustained overhead.

Therefore, as it shows in Table 8.3, the protocol maintains linear scalability relative to the number of participants per negotiation round and incurs only a single quadratic overhead due to the inherent requirements of BFT consensus. Given typical consortium scales in industrial contexts (tens, rather than thousands, of organizations), these complexity measures are sufficiently low and practically insignificant compared to routine industrial IoT traffic.

8.2.3.1 Negotiation Process

To enable secure and policy-compliant inter-organizational cooperation, this framework introduces a decentralized, iterative negotiation protocol. Its design supports autonomous decision-making while promoting convergence toward mutually acceptable policies.

Let each participating entity be modeled as a PE. At step $t = 0$, each PE generates a proposal $P^{(0)}$ composed of access parameters such as permission level, authentication method, temporal validity, and usage quotas. These values are derived from internal constraints and policies, without revealing sensitive internal logic.

Upon receiving a proposal $P^{(t)}$, each PE computes the following:

- The *Utility* score $U(P^{(t)}) = \sum_{i=1}^{n} w_i \cdot f_i(x_i)$, where w_i are preference weights and $f_i(x_i)$ are normalized objective satisfaction scores.
- The *Risk* score $R(P^{(t)}) = \frac{\text{ThreatLikelihood}\cdot\text{Impact}}{\text{MitigationFactor}}$, quantifying the expected harm of accepting the proposal.
- The *Compliance* score $C(P^{(t)}) \in [0, 1]$, reflecting alignment with regulatory and governance requirements.

A proposal is accepted if:

$$U(P^{(t)}) \geq \tau_u \quad \wedge \quad R(P^{(t)}) \leq \tau_r \quad \wedge \quad C(P^{(t)}) \geq \tau_c$$

where τ_u, τ_r, τ_c are threshold values for utility, risk, and compliance respectively, defined per entity. Each entity defines its own thresholds based on internal security posture and regulatory exposure. For example, a highly regulated pharmaceutical company might require $\tau_c = 0.95$ for compliance, tolerate $\tau_r = 0.2$ for risk, and expect $\tau_u = 0.7$ for utility. In contrast, a logistics provider may prioritize availability and accept slightly higher risks in exchange for operational flexibility.

If the proposal fails to satisfy acceptance conditions, the receiving PE generates a counter-proposal $P^{(t+1)}$ by modifying parameters that reduce risk or improve utility. The *Refine* function is a local decision-making mechanism executed by each PE when a received proposal does not meet predefined acceptance thresholds. Its role is to systematically generate a new, counter-aligned proposal $P^{(t+1)}$, derived from the previous version $P^{(t)}$, by modifying its parameters to improve alignment with internal objectives and constraints. These modifications may include tightening access

permissions to reduce security risks, increasing authentication strength, shortening temporal validity to limit exposure, or relaxing non-critical operational constraints to increase overall acceptability. The refinement process relies on predefined organizational preferences, risk tolerance levels, and compliance priorities. While each PE operates independently and without disclosing its internal utility function, the refinement aims to converge toward mutually acceptable trade-offs. The function is deterministic and context-aware, allowing organizations to iteratively adjust their stance while respecting the negotiation protocol's privacy and scalability constraints.

The negotiation continues iteratively:

$$P^{(t+1)} = \text{Refine}(P^{(t)})$$

until convergence is achieved or a termination condition is met (e.g., time-out or maximum iterations).

By anchoring each proposal and response on a blockchain, the system ensures tamper-proof, auditable negotiation logs. This approach enables privacy-preserving, scalable, and trust-aware policy alignment without requiring full policy disclosure or central coordination.

Security Guarantees in Negotiation. Each negotiation message is signed, timestamped, and validated through mutual TLS. Replay attacks and message injection are prevented through freshness and authentication checks. Blockchain anchoring ensures proposals cannot be retroactively altered.

To operationalize this process in a decentralized and verifiable manner, the entire negotiation protocol is designed to be executed as a smart contract deployed on the blockchain. This smart contract functions as a distributed state machine that governs the interaction between PPEs of the participating organizations. Each PE interacts with the smart contract by submitting proposals, performing utility and risk assessments, and issuing counter-proposals. The contract enforces timing constraints, manages the negotiation state, and finalizes the agreement once consensus is reached. Importantly, this architecture keeps IIoT devices out of the negotiation loop, relying instead on organizational PEs or intermediary nodes to handle all blockchain interactions ensuring compatibility with resource-constrained environments.

This process ensures privacy-preserving collaboration by maintaining the integrity of each organization's local policies while enabling secure, transparent, and auditable cross-company interactions. By leveraging the blockchain, the framework provides a scalable and robust solution to the challenges of distributed policy management and negotiation.

8.3 Use Case: Policy Negotiation Workflow

The proposed methodology is applied in the context of a multinational supply chain network, involving various stakeholders such as manufacturers, suppliers, logistics providers, distributors, and retailers. Each organization operates under distinct security standards, legal frameworks, and internal policies. These differences create challenges for secure and compliant data sharing.

Let us consider the scenario illustrated in Figure 8.2. Company A restricts access to its machine data to devices within its office IP range. Company B allows machine access only to operators directly involved in its operations. Company C treats data related to its product X as confidential and non-shareable.

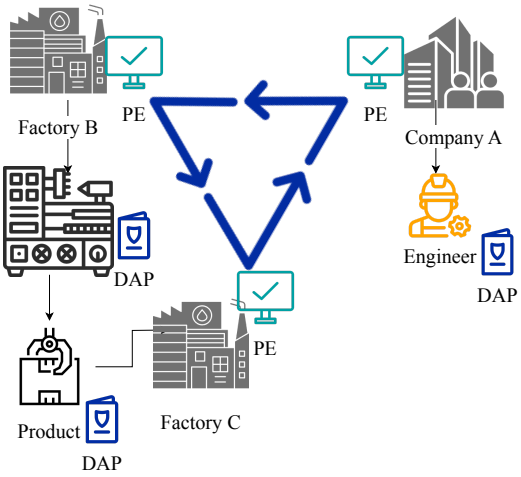


Figure 8.2: Use Case Scenario.

8.3.1 Context Setup

As shown in Figure 8.2, each company operates a Policy Engine (PE) and maintains DPPs encapsulating their DAPs. A blockchain provides immutable record-keeping of finalized agreements. For example:

- **Company A** requires IP-based restrictions and multifactor authentication (MFA) for engineers accessing external data;
- **Company B** restricts data based on operator roles, exposing only certain diagnostic parameters from its machines;
- **Company C** protects product X-related data, insisting on anonymization before sharing.

8.3.2 Initiating Negotiation and Policy Retrieval

The following narrative provides a step-by-step walkthrough of the collaboration scenario depicted in Figure 8.3, detailing how each organization initiates, retrieves, evaluates, and negotiates policy agreements within the proposed framework. Company A initiates a collaboration request, seeking access to certain operational and product-related data. This "request for consensus" message propagates to Company B and, subsequently, to Company C. All parties now must align their policies.

Each PE retrieves the current policies from the relevant DPP instances:

- Company A's PE fetches its DAPs from the DPP engineer instance, confirming that IP and MFA conditions apply.

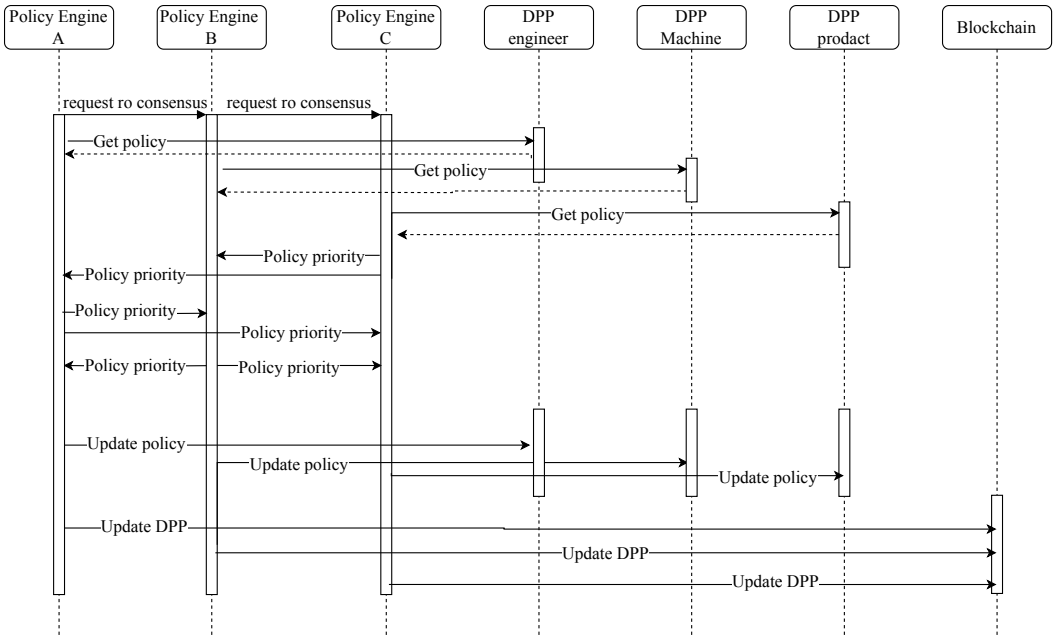


Figure 8.3: Sequence Diagram of the Negotiation and Policy Update Process.

- Company B's PE queries the DPP machine instance to review operational data-sharing rules.
- Company C's PE checks the DPP product instance to enforce anonymization of product X-related data.

The "Get policy" messages in the sequence diagram ensure that all negotiation partners start from an up-to-date, accurate baseline.

8.3.3 Policy Priority Exchanges and Automated Negotiation

After retrieving the policies, the PEs exchange multiple "Policy priority" messages, representing the iterative negotiation process. Each PE evaluates proposals using:

- **Utility and Risk Assessment:** Determining whether the proposed terms meet operational needs while minimizing security risks.
- **Compliance Checks:** Ensuring adherence to legal, regulatory, and corporate governance mandates.

During this iterative phase:

- Company A may reconfigure its IP filtering policies to permit external access under stricter authentication requirements, such as mandatory MFA.
- Company B agrees to reveal limited diagnostic parameters while withholding sensitive operational data.
- Company C insists on data anonymization procedures to protect product X information.

These back-and-forth exchanges continue until the PEs find a mutually acceptable set of policies or a predefined timeout occurs. The sequence diagram's repeated "Policy priority" messages reflect these negotiation loops.

8.3.4 Updating Policies and Committing to Blockchain

Once consensus is reached, each PE sends "Update policy" commands to its local DAPs, ensuring that the PEPs reflect the newly agreed-upon terms. For example:

- Company A's updated policy now grants Company B's engineers access to certain machine data under MFA and IP constraints.
- Company B enforces a filtered subset of diagnostic parameters for Company A and C.
- Company C finalizes the anonymization rules to ensure no identifying product X data leaves its domain.

The final step is recording these updated policies in the DPPs and pushing the changes to the blockchain. While no raw data is stored on-chain, the blockchain's immutable ledger preserves the final policy state, enabling:

- An irrefutable audit trail of the negotiation.
- Future verification of compliance and resolution of disputes.
- Maintenance of trust among all parties, as no unilateral changes can be made without renegotiation.

8.3.5 Post-Agreement Operations and Security Rationale

With policies aligned, subsequent data requests and sharing activities proceed smoothly. Data is exchanged off-chain and encrypted on-demand using the requester's public key, ensuring that even if certain keys are compromised, the blockchain record remains secure and only policy conditions (not raw data) are at risk.

Continuous monitoring detects any non-compliance, triggering alerts or new negotiations if necessary. The combination of secure communication, cryptographic protections, and blockchain-based immutability upholds:

- **Confidentiality:** Sensitive data never resides on-chain, reducing exposure.
- **Integrity and Non-Repudiation:** Immutable records verify that all parties adhere to agreed terms.
- **Dynamic Adaptability:** Policies can be renegotiated as business requirements evolve, with each update permanently logged on the blockchain.

This scenario showcases how automated negotiation, supported by robust policy retrieval, iterative priority-based adjustments, and reliable blockchain records, can foster secure, trusted, and regulation-compliant data sharing in complex multinational supply chains.

8.4 Security Guarantees and Discussion

This section evaluates how the architectural mechanisms introduced in Section 8.2.1 translate into practical security guarantees under realistic operational and adversarial conditions. Rather than re-stating implementation details, the focus here is on interpreting the security properties they enable, alongside their limitations and underlying assumptions.

8.4.1 Achieved Guarantees in Context

The integration of cryptographic enforcement, decentralized policy validation, and immutable blockchain logging leads to a number of foundational guarantees:

- **Confidentiality:** Sensitive policy content remains encrypted and off-chain. Access is mediated through authenticated channels and governed by strict Zero Trust enforcement.
- **Integrity and Authenticity:** All policy records are digitally signed and immutably committed to the blockchain. Any tampering or forgery attempts are immediately detectable.
- **Non-Repudiation:** Every negotiation action is logged with cryptographic signatures, providing a verifiable audit trail and ensuring participant accountability.
- **Availability and Resilience:** A replicated, distributed ledger ensures that policy data remains accessible even in the presence of node failures or partial network outages.
- **Regulatory Compliance:** Compliance checks are embedded directly into the policy evaluation process and enforced during negotiation, with results permanently logged for auditing.

These guarantees are not derived from abstract theoretical models, but rather from concrete mechanisms embedded in the system's architecture and protocol logic.

8.4.2 Discussion of Remaining Risks and Assumptions

These guarantees hold under several key assumptions, which must be maintained operationally:

- **Cryptographic Soundness:** The design presumes the security of underlying primitives such as encryption schemes, digital signatures, and hash functions. Attacks on these components (e.g., side channels or faulty key storage) are considered out of scope.
- **Authenticated Communication:** Secure communication channels (e.g., mutual TLS with certificate pinning) between nodes are assumed. A breach at this layer could compromise enforcement and trust boundaries.

- **Threshold Trust in BFT Consensus:** The blockchain's consensus protocol requires that fewer than one-third of participating nodes are malicious. Breaching this assumption may impact liveness and, in extreme cases, consensus safety.
- **Operational Diligence:** Participating organizations must maintain secure key management, policy evaluation modules, and node uptime. Misconfigurations or human error could bypass intended security safeguards.
- **Privacy-Utility Considerations:** The architecture emphasizes privacy through cryptographic isolation and off-chain data storage. While this limits visibility into policy content, it ensures regulatory compliance and data confidentiality. Real-time enforcement is handled at the edge via pre-evaluated policies and cached decisions, mitigating latency introduced by encrypted data access. Network administrators retain visibility into metadata flows, logs, and audit trails without exposing sensitive content. A full quantitative assessment of tradeoffs (e.g., latency vs. confidentiality) is planned as part of future performance evaluation.

In addition, we assume initial trust is established via pre-distributed cryptographic identities tied to organizational certificates, managed through a permissioned blockchain framework. These credentials enable secure bootstrapping and authenticated communication (e.g., mutual TLS). The system leverages BFT consensus and thus tolerates up to $f < n/3$ malicious nodes in a network of n participants. While not directly addressed in this work, known blockchain-specific threats such as eclipse attacks, and physical tampering of IIoT endpoints, remain significant vectors and will be considered in future extensions of the security model.

Finally, the use of blockchain infrastructure introduces non-negligible performance and resource costs. These include latency introduced by consensus, increased bandwidth usage, and long-term storage overhead. While acceptable in many industrial settings, such overhead must be considered carefully in latency-sensitive or resource-constrained IIoT deployments.

8.5 Conclusion

This chapter presented the "Distributed ZTA based on Policy Negotiation Secured by DPP in Blockchain", an innovative approach designed explicitly for securing distributed, interconnected IIoT environments. By uniquely integrating blockchain technology with Digital Product Passport mechanisms, the proposed architecture enables decentralized, secure, and fully verifiable policy negotiation and enforcement across multiple independently managed industrial domains.

The key advantage of this architecture lies in its ability to provide transparent, tamper-resistant policy management and secure interoperability among diverse stakeholders without reliance on centralized trust authorities. This approach significantly advances previous decentralized trust solutions by ensuring real-time adaptability, improved operational transparency, and robust security against policy manipulation.

However, despite these advancements, certain challenges remain. Implementing blockchain-based policy negotiations can introduce additional complexity and computational overhead, particularly in resource-constrained IIoT environments. Additionally, maintaining high performance and scalability as networks expand remains a critical operational consideration, requiring careful optimization and ongoing research.

In conclusion, this chapter contributes a scalable, transparent, and adaptable trust management system that strengthens the overall IIoT security framework developed throughout the thesis. While the architecture offers a holistic approach to secure collaboration in distributed environments, its full implementation lies beyond the scope of this work due to resource constraints. As such, it remains a strong candidate for future industrial deployment and cross-organizational collaboration, with the next phase of research focusing on integration challenges, performance evaluation, and real-world adoption.

Chapter 9

Conclusion and Future Research Directions

In this thesis, we investigated how a combination of ZTA, blockchain-based integrity frameworks, and context-aware anomaly detection can can enable adaptive and intelligent security mechanisms for critical IIoT deployments. We first underscored the complexity of industrial networks through a smart factory use-case analysis. Next, we introduced mechanisms for cross-domain identity management, emphasizing federated authentication and digital wallets. We proposed anomaly detection solutions that integrate neural networks and graph-based clustering to identify unknown threats effectively. We then introduced a lightweight blockchain mechanism specifically adapted for industrial scale, focusing on auditing and DPPs. Finally, we synthesized these findings into a Dynamic and then a Distributed Zero Trust Architecture, demonstrating how multi-stakeholder collaboration can be achieved securely in a truly decentralized manner. The main achievements can be summarized as follows:

- **Foundational Identity Management and Authentication:** A thorough exploration of digital wallet identity schemes and cross-domain authentication protocols led to an architecture that unifies identity verification processes in IIoT environments. The proposed methods reduce bottlenecks and latency commonly encountered in centralized identity management systems, meeting industrial requirements for low-overhead operation and continuous trust verification.
- **Context-Aware Anomaly Detection:** By combining graph-based autoencoders, Linear Discriminant Analysis (LDA), and community detection, the research highlights how IIoT devices' contextual relationships can be leveraged to detect and classify anomalies in real time. Empirical evaluations in both simulation and testbeds demonstrate not only the robustness of these approaches against a range of cyberattacks, but also the adaptability to dynamically shifting process parameters in industrial environments.
- **Lightweight Blockchain Solutions:** Recognizing the limited computational resources of certain IIoT nodes, the thesis proposed and validated lightweight

blockchain designs tailored to shop-floor environments. These solutions emphasize fast consensus, minimized storage footprints, and selective confidentiality thereby preserving key blockchain benefits such as immutability and decentralization, while introducing minimal performance trade-offs.

- **Dynamic and Distributed Zero Trust Architectures:** Building on fundamental Zero Trust Architecture (ZTA) concepts, the work introduces a dynamic Zero Trust framework that iteratively refines trust scores and policies based on ongoing risk assessments, device context, and anomaly alerts. A distributed version of this ZTA leverages blockchain to negotiate and enforce security policies across domain boundaries, reducing single points of failure and fostering collaborative trust management among industrial stakeholders.
- **Holistic Security Perspective:** By tightly coupling identity-centric controls, real-time anomaly detection, and decentralized trust, the thesis offers a multidimensional security approach that can scale with the evolving complexity of Industry 4.0. The integrated viewpoint ensures that mitigating actions (such as blocking compromised devices) are triggered quickly and precisely, balancing security rigor with operational continuity.

9.1 Revisiting and Answering the Research Questions

In this section, we revisit the research questions initially outlined in Chapter 1 (Section 1.5) and concisely summarize how each question was addressed through the methodologies and approaches developed throughout this thesis.

RQ1: What are the challenges in integrating blockchain and Zero Trust principles into a cohesive security framework for IIoT

This thesis identified several integration challenges, including balancing security robustness with resource constraints (Chapter 6), managing trust relationships dynamically across distributed domains (Chapter 4 and Chapter 8), and handling the computational and latency overhead associated with blockchain and Zero Trust mechanisms (Chapter 7). Solutions proposed include a lightweight blockchain implementation tailored for IIoT environments and the adoption of context-aware dynamic policies within a decentralized Zero Trust framework.

RQ2: How can hybrid and context-aware anomaly detection methods improve the real-time identification and assessment of sophisticated security threats within dynamic IIoT networks?

Chapters 5 and 7 demonstrated that hybrid anomaly detection methods combining deep-learning autoencoders (AE) and linear discriminant analysis (LDA), alongside context-aware community-based graph detection, significantly improve real-time threat identification and assessment. These approaches effectively distinguish genuine threats from benign anomalies, achieving high accuracy and reducing false positives, thus enhancing the security posture in dynamic IIoT settings.

RQ3: How can scalable and secure identity management be achieved in distributed IIoT networks?

Secure and scalable identity management was addressed through a novel digital wallet approach (Chapter 4), providing decentralized, secure management of identities across multiple industrial domains. This solution leveraged blockchain-backed identity validation mechanisms, enabling efficient cross-domain authentication without central bottlenecks, ensuring scalability, interoperability, and resilience to identity-related threats.

RQ4: What are the limitations of traditional Zero Trust architectures in dynamic IIoT environments, and how can they be adapted for real-time security?

Limitations identified in traditional Zero Trust models included static policy constraints, latency issues in real-time environments, and the challenges of continuous trust verification in dynamic contexts (Chapters 7 and 8). To overcome these limitations, a dynamic Zero Trust framework was developed, incorporating real-time risk scoring, continuous authentication, and adaptive policies informed by anomaly detection results, thus enabling real-time security adaptation in industrial networks.

RQ5: How can blockchain address IIoT challenges of scalability, privacy, and tamper-proofing?

Blockchain solutions presented in Chapter 6, specifically the Shopfloor and Lightweight Blockchain approaches, successfully addressed these challenges. The lightweight blockchain framework achieved reduced computational complexity and enhanced transaction throughput, tailored explicitly for resource-constrained IIoT devices. Simultaneously, privacy-preserving techniques embedded in the blockchain ensured robust tamper-proofing and confidentiality, thus meeting IIoT operational demands.

RQ6: How can identity management, blockchain, and anomaly detection be integrated into a cohesive Distributed ZTA framework for securing IIoT networks?

The integration of these elements into a cohesive security framework was detailed in Chapters 7 and 8. The Distributed ZTA framework combines digital wallet-based identity management for scalable cross-domain authentication, lightweight blockchain to manage decentralized trust, and advanced context-aware anomaly detection for proactive threat assessment. The resulting integrated framework demonstrated robust security enforcement, scalability, and adaptability suitable for distributed, heterogeneous, and dynamic industrial environments.

9.2 Short-Term Perspectives

The proposed multi-layered security framework comprehensively addresses several fundamental challenges in securing distributed IIoT systems. Throughout the research and prototyping process, three short-term critical challenges emerged as crucial factors shaping the architectural direction of this thesis: (1) resource constraints devices, (2) heterogeneous security requirements across different industrial sectors, and (3) the continuously evolving and unpredictable nature of the threat landscape. Each of these challenges required distinct solutions, leading to the development of three complementary security layers: a lightweight blockchain infrastructure, a modular identity and access control design, and a dynamic, risk-aware ZTA framework. The following subsections describe these challenges in depth and explain how they guided the layered contributions in this thesis.

9.2.1 Resource-Constrained Devices

A central technical challenge addressed in this thesis is the integration of security mechanisms in legacy and low-power industrial devices that lack the computational, memory, or energy capacity to execute standard cryptographic protocols. These constraints are especially problematic in the context of blockchain-based models, which traditionally depend on computationally intensive operations such as consensus, signing, and hashing.

In response, the thesis introduced two key mechanisms tailored for constrained environments: a lightweight attribute verification protocol and a lightweight blockchain architecture (Chapter 6.4). The attribute verification protocol enables devices to prove possession of authorized attributes using minimal cryptographic operations, avoiding the need for heavy signature checks or zero-knowledge proofs. In parallel, the blockchain architecture decouples core functions and distributes them across a hierarchy of roles. This allows limited devices to participate in the system without performing costly computations, while relying on higher-tier nodes for validation and consensus.

The research challenge that emerged during this work is the need to adapt cryptographic protocols themselves to operate reliably on constrained hardware while remaining compatible with distributed infrastructures. Adapting cryptography in this context involves more than using faster algorithms; it requires rethinking protocol flows, key management, and secure data handling to minimize state, reduce packet size, and tolerate hardware limitations.

The approach proposed in this thesis involves leveraging device classification and provisioning profiles to dynamically determine cryptographic responsibilities. By tailoring security tasks to device capabilities, the system preserves end-to-end trust without uniformly imposing heavy cryptographic loads. Lightweight devices verify their eligibility through efficient attribute proofs, while heavier blockchain roles handle consensus and auditability.

What makes this approach original is the combination of lightweight cryptographic delegation with attribute-based verification, all embedded within a scalable, distributed architecture. This ensures that constrained devices are not excluded from secure participation, but are instead integrated through role-aware,

resource-sensitive security mechanisms that maintain the integrity of the overall trust infrastructure.

9.2.2 Heterogeneous Security Requirements

Industrial systems operate in vastly different contexts—ranging from smart manufacturing to critical infrastructure and healthcare, each with specific security goals and compliance obligations. This diversity results in heterogeneous requirements for confidentiality, availability, data integrity, and accountability, often dictated by sectoral regulations such as IEC 62443, ISO 27001, or GDPR.

The thesis addressed this heterogeneity by introducing a modular, policy-based security framework (Chapters 4, 7, and 8) in which devices and domains are treated as individually classifiable entities, each associated with its own policy context. This allowed different access and trust configurations to coexist within the same architecture.

The underlying research challenge, however, is not only about modularity, but about how to formally link high-level regulatory requirements with runtime policy enforcement. While the framework supports per-device customization, the translation of compliance constraints into enforceable logic remains a complex and underexplored problem, especially in dynamic, multi-tenant environments.

This thesis proposes a partial solution by embedding relevant compliance attributes within digital wallets and using them to influence policy evaluation and decision-making. This allows identity credentials to carry contextual information (such as sector classification or device criticality) that can be used to align enforcement with external requirements.

The originality of this approach lies in the integration of regulatory semantics into the identity layer. Rather than configuring policies manually for each domain, the system uses credential metadata to drive security adaptation, creating a dynamic bridge between operational enforcement and formal compliance categories.

9.2.3 Dynamic Threat Landscape

The evolving nature of cybersecurity threats presents a constant challenge for IIoT environments, where static protection models quickly become outdated. The rise of zero-day exploits, supply chain attacks, and behaviorally evasive malware has demonstrated that predefined access control rules and traditional detection systems are insufficient to protect industrial networks.

To address this, the thesis introduced a Dynamic Zero Trust Architecture (Chapter 7) that integrates real-time anomaly detection with policy enforcement. Anomaly detection methods based on temporal patterns and graph-based machine learning (Chapter 5) were used to recognize suspicious behavior and adjust device-level policies accordingly.

The research challenge is how to ensure this adaptation happens continuously, accurately, and safely. Reacting to emerging threats requires detection systems to generalize beyond known attack signatures and requires policy layers to react without destabilizing operations or opening new vulnerabilities.

This thesis approaches the problem by linking anomaly classification to the state transitions of a finite-state machine-based policy engine. This allows the system to

respond to threats in a controlled and predictable way, adjusting the trust level of devices and triggering appropriate mitigation steps.

The originality of this contribution lies in the integration of detection and enforcement into a single feedback loop. Unlike traditional architectures where policy and monitoring are decoupled, this system continuously evaluates trust and adapts policy using live context. This approach ensures not only more timely responses, but also tighter coupling between what is observed and what is enforced.

9.3 Long-Term Research Challenges and Scientific Outlook

Challenge 1: Trust Coordination Without Central Policy Anchors

This thesis introduced a decentralized policy negotiation protocol that enables autonomous organizations in IIoT ecosystems to exchange and evaluate access policies without relying on a central authority or pre-defined trust anchor (Chapter 8). The protocol allows independently managed actors (such as suppliers, operators, and regulators) to negotiate access rights based on verifiable claims and contextual trust signals. This represents a significant step forward in enabling secure collaboration across administrative boundaries while preserving autonomy and policy ownership.

However, a critical challenge remains unresolved: how can such policy negotiations remain effective and verifiable when participants do not share a common vocabulary, classification schema, or semantic model for roles, attributes, and device categories? In real-world deployments, stakeholders define trust and security policies based on different operational, regulatory, or business contexts. Even if the protocol enables the exchange of policies and claims, misalignment in meaning can lead to incorrect trust assumptions or rejected negotiations, not due to security conflicts, but due to semantic mismatches.

The long-term research problem is therefore not about designing the negotiation protocol itself (this thesis solution) but about enabling semantic interoperability within that protocol. In other words, how can two IIoT entities negotiate trust when they describe their requirements and capabilities using different terms, structures, and implicit assumptions?

Addressing this problem requires a rethinking of how policy knowledge is represented, translated, and reconciled across domains. A future research direction would explore the use of self-descriptive policy representations, where each domain's security policies include embedded semantic metadata. This would allow policies to be interpreted contextually, supporting negotiation even in the absence of a pre-agreed vocabulary. Such representations could build on semantic web technologies or domain-specific ontologies, but would need to be designed for real-time, privacy-preserving negotiation environments.

To ensure trustworthiness, future work could also explore the use of zero-knowledge proof mechanisms within the negotiation protocol. While this thesis does not implement such cryptographic techniques, it proposes their potential to preserve the privacy of internal policy structures and prevent unauthorized manipulation of the negotiation process. Zero-knowledge techniques could allow a party to prove that it complies with an externally defined policy—without revealing its own access

logic or credentials. Furthermore, these proofs could enforce strict bounds on what is verified, reducing the risk of policy injection or scope creep during negotiation.

Challenge 2: Machine-Led Security Reasoning in Dynamic Contexts

Industrial systems are increasingly exposed to complex, evolving threats that exploit structural and behavioral vulnerabilities in unpredictable ways. Zero-day attacks, lateral movements, firmware manipulations, and multi-stage exploits often defy traditional defense models. In such environments, security mechanisms must go beyond static rules or known patterns—they must detect the unfamiliar, infer the unexpected, and respond intelligently.

This thesis addressed this challenge through two complementary anomaly detection methods (Chapter 5). The first approach used an autoencoder to model the normal behavior of network traffic. By learning typical communication patterns, the model was able to flag deviations—especially those caused by novel or stealthy attacks that had never been seen during training. This makes it well-suited to detecting zero-day threats without relying on labeled data. The second method modeled the IIoT network as a graph and applied community detection and graph-based metrics to identify suspicious relational patterns and contextual anomalies, such as unexpected interactions between devices or shifts in community structure. Together, these approaches allow the system to detect both low-level behavioral deviations and higher-order structural anomalies.

While these methods represent an important advancement beyond traditional detection, the next research challenge is to develop systems that can go further: not only identifying that something abnormal is happening, but also reasoning about why it might be happening, how it could evolve, and what action should be taken. The problem is no longer just detection, it is autonomous, machine-led security reasoning in dynamic and partially observable environments.

The long-term research goal is to design IIoT systems capable of generating structured, interpretable explanations for unfamiliar behaviors, simulating possible attack progressions, and proactively adjusting policies or triggering mitigations based on inferred intent. For example, a system could detect a subtle change in device communication, infer that it resembles early-stage lateral movement, and adjust access controls preemptively—before the attacker escalates privileges.

This direction builds on the existing contributions of this thesis by proposing the integration of generative models and causal inference into the detection and decision loop. Generative models could simulate possible threat variations or attack paths, extending detection coverage into hypothetical spaces that the system has not observed by game theory methods. Causal inference techniques would help explain which behaviors contribute to anomalies and estimate their likely origins or consequences. These capabilities would not replace anomaly detection, but rather enhance its output with interpretability, anticipation, and actionability.

Feasibility could be evaluated through simulation environments containing realistic IIoT topologies and multi-stage attack scenarios. Metrics would include not only detection accuracy, but also explanation quality, response time, and the effectiveness of automated mitigations. Reasoning outputs would need to be auditable and verifiable to ensure they support both human oversight and policy adaptation.

9.4 Concluding Remarks

This thesis reinforces the need for a cohesive, automated, and multi-layered security strategy tailored to the operational realities of modern IIoT environments. By integrating identity management, dynamic policy evaluation, context-aware anomaly detection, and decentralized trust mechanisms, the proposed framework addresses the inherent complexity, heterogeneity, and evolving threat landscape of industrial systems.

Rather than relying on static controls or perimeter-based assumptions, this work emphasizes the shift toward adaptive and risk-informed decision-making. It demonstrates how Zero Trust principles can be operationalized in a dynamic and context-sensitive manner without central coordination while maintaining low overhead suitable for industrial deployments.

Looking ahead, this contribution lays the foundation for scalable, autonomous security infrastructures capable of continuous adaptation. Future research can build upon this by refining the underlying trust negotiation logic, optimizing lightweight cryptographic protocols, and extending real-time contextual integration. These directions will be critical to advancing secure, resilient, and self-regulating cybersecurity architectures that align with the demands of Industry 4.0.

List of Figures

List of Tables

Acronyms

AB AdaBoost. 24

ABAC Attribute-Based Access Control. 19, 22, 23, 189

AE Autoencoder. 66, 71, 75–79, 96, 98

AUC Area Under the Curve. 91

AUROC Area Under the ROC Curve. 77–79, 96

BFT Byzantine Fault Tolerance. 31, 32, 102, 104, 114, 115, 117, 119, 120, 122, 123, 126, 162, 180, 181, 196–199

BoL Beginning of Life. 14, 189

CA Certificate Authority. 52, 56, 57, 60

CapBAC Capability-Based Access Control. 18, 189

CART Classification and Regression Trees. 90

CBC Consistent Broadcast. 31

CN Committee Node. 115–119, 123, 124

CNN Convolutional Neural Network. 25

DAG Directed Acyclic Graph. 31, 32, 102, 114–116, 118–120, 122, 123, 125, 126, 180, 181, 196–199

DAP Data Access Passport. 155, 156, 160, 164, 166

DDoS Distributed Denial-of-Service. 120, 121

DID Decentralized Identifier. 16, 47, 189

DoS Denial-of-Service. 13, 41, 42, 78, 103, 113, 120, 121, 188, 196

DPP Digital Product Passports. 10, 13, 32–34, 154–156, 160, 164, 166, 168, 171, 179, 190

DT Decision Tree. 24

ECC Elliptic-Curve Cryptography. 53

ECN Edge Computing Node. 115, 116, 122–124

EoL End of Life. 14, 33, 189

EoP Elevation of Privilege. 41

EUF-CMA Existential Unforgeability under Chosen Message Attack. 51, 62

FN Full Node. 105, 111, 114

FSM Finite State Machine. 133–135, 138, 139

GCN Graph Convolutional Network. 86, 87

GNB Gaussian Naive Bayes. 24

GNN Graph Neural Network. 26, 70, 83, 86, 89, 96, 189

GRU Gated Recurrent Unit. 25

HMAC Hash-Based Message Authentication Code. 41

HSM Hardware Security Module. 45, 46, 105, 106, 196, 197

IAM Identity and Access Management. 10, 13, 23, 44, 190

ICPS Cyber-Physical Systems. 21

IdM Identity Management. iii, 15, 16, 189

IDS Intrusion Detection System. 25, 78, 79

IIoT Industrial Internet of Things. iii, 1–5, 8, 10–18, 23, 24, 26–31, 34, 35, 37–42, 44–49, 52, 53, 56, 63, 66–70, 86, 98, 102, 103, 105, 106, 111, 113–115, 120–122, 126, 127, 130, 131, 146, 151–153, 168, 169, 172–174, 178, 179, 181, 187, 188, 190, 193, 196

IMS Identity Management Service. 46–49

INS Installer Signature. 107, 108, 112

IoT Internet of Things. 10, 11, 18, 26, 27, 30–32, 34, 37–40, 42, 85, 89, 94, 97, 98, 104, 125, 126, 188

KNN K-Nearest Neighbor. 24

LDA Linear Discriminant Analysis. 66, 71, 73–79, 96, 98

LN Local Nodes. 105–110, 112

LPA Label Propagation Algorithm. 85, 194

LSTM Long Short-Term Memory. 25, 27

MFA Multifactor Authentication. 41, 164–166

MitM Man-in-the-Middle. 13, 120, 188

MN Middle Node. 105–108, 110, 112–114

MoL Middle of Life. 14, 189

MSE Mean Squared Error. 73, 74, 77, 78

OWASP Open Worldwide Application Security Project. 40

PA Policy Administration. 19, 132, 135, 140, 146

PBFT Practical Byzantine Fault Tolerance. 28, 103, 105, 111, 113, 114, 196

PCAP Packet Capture. 71, 72

PDP Policy Decision Point. 17, 18, 20, 154, 189

PE Policy Engine. 19, 132–135, 138–140, 146, 155, 156, 158, 160, 162–166

PEP Policy Enforcement Points. 17–20, 132, 133, 146, 155, 160, 166, 189

PeW Proof of Equivalent Work. 32

PKI Public-Key Infrastructure. 15, 41, 54, 56, 189

PLC Programmable Logic Controller. 11, 38, 39

PoA Proof of Authority. 31, 103

POC Power Consumption. 107, 108, 112, 113

PoH Proof of History. 124

PoS Proof of Stake. 113, 124, 126

PoW Proof of Work. 28, 31, 32, 103, 113, 124–126

PUF Physically Unclonable Function. 53, 189

RBAC Role-Based Access Control. 16, 18, 22, 23, 42, 189

RBC Reliable Broadcast. 31

ReLU Rectified Linear Unit. 74, 87, 88

RF Random Forest. 24

RN Regular Node. 115, 116, 121, 123, 124

RNN Recurrent Neural Networ. 25, 26

ROC Receiver Operating Characteristic. 76, 91, 93

SCADA Supervisory Control and Data Acquisition. 11, 38–42, 54

SDN Software-Defined Networking. 38

SEC Logical Network Sector. 107, 108, 112

SGD Stochastic Gradient Descent. 24

SHAP SHapley Additive exPlanations. 72, 194, 195

SLA Service-Level Agreement. 12

SSI Self-Sovereign Identity. 16, 189

SSO Single Sign-On. 16

SVM Support Vector Machine. 24, 25, 74

TBAC Trust-Based Access Control. 21

TGN Temporal Graph Network. 26

TLS Transport Layer Security. 44, 53–58, 62

TPM Trusted Platform Module. 45–47

TPS Transactions Processed per Second. 122–126, 180

TRA Transmission Pattern. 107, 108, 112

TVN Transaction Verification Nodes. 115–117, 120

VAE Variational Autoencoder. 24, 25

VC Verifiable Credential. 47

XAI Explainable AI. 71

ZTA Zero Trust Architecture. iii, 2–5, 10, 13, 19–23, 34, 130, 131, 135, 143, 146, 147, 152, 154, 155, 168, 171, 173, 187, 188, 190

Appendix A

Résumé en français

Introduction

L'Internet industriel des objets (IIoT) accroît l'automatisation et le suivi en temps réel dans de nombreux secteurs, mais élargit fortement la surface d'attaque. Les défenses périmétriques ne tiennent plus dans des usines multi-domaines connectées à Internet. Les dispositifs à ressources limitées ne peuvent supporter des mécanismes cryptographiques lourds ; la confiance centralisée devient un goulot d'étranglement et un point de défaillance unique ; et les opérations temps réel ne tolèrent pas la latence induite par la sécurité. La thèse plaide pour une sécurité adaptative et sensible au contexte, à la fois scalable et réactive en temps réel.

La ZTA renforce l'assurance via le principe « ne jamais faire confiance, toujours vérifier », mais ses déploiements restent souvent statiques et coûteux pour l'IIoT. La blockchain apporte décentralisation, intégrité et auditabilité pour la confiance inter-parties, mais les conceptions naïves induisent des surcoûts de calcul/stockage et de la latence. Ce travail propose une intégration pratique : une ZTA dynamique et contextuelle combinée à une blockchain légère adaptée à l'IIoT.

Objectifs.

- Authentification inter-domaines.
- Politiques ZTA adaptatives au contexte.
- Confiance décentralisée efficace via une blockchain légère.
- Détection d'anomalies contextuelle avec faible taux de faux positifs.
- Évaluation complète dans des scénarios réalistes.

Questions de recherche. Intégrer blockchain + ZTA ; concevoir une détection hybride et contextuelle ; assurer une gestion d'identités scalable ; adapter la ZTA au temps réel de l'IIoT ; étudier le rôle de la blockchain pour l'évolutivité, la confidentialité et l'inaltérabilité ; unifier le tout dans une ZTA distribuée.

Contributions. Cette thèse apporte des avancées significatives dans la sécurisation des environnements IIoT distribués. Les contributions spécifiques sont les suivantes :

1. **Gestion Intégrée de l'Identité pour l'IIoT :** Nous introduisons un mécanisme d'authentification inter-domaines exploitant les portefeuilles numériques pour intégrer et gérer en toute sécurité les appareils industriels.
2. **Détection d'Anomalies Sensible au Contexte :** Nous développons des stratégies avancées de détection d'anomalies qui intègrent des architectures de réseaux de neurones avec des techniques de détection de communautés basées sur des graphes.
3. **Blockchain Légère et Passeports Numériques de Produits :** Nous proposons un framework de blockchain légère adapté aux applications industrielles et intégrons les passeports numériques de produits pour la gestion du cycle de vie des actifs.
4. **Architecture Zero Trust Dynamique :** Nous concevons une Architecture Zero Trust extensible qui évalue continuellement les niveaux de menace en utilisant des évaluations de risque en temps réel.
5. **Paradigme de Contrôle d'Accès Entièrement Distribué :** Nous étendons le framework ZTA en introduisant un protocole de négociation de politiques décentralisé pour la collaboration multi-acteurs.

Structure. Le document progresse de l'état de l'art et de l'analyse de scénario vers l'identité, la détection d'anomalies, une blockchain légère, la ZTA dynamique, son extension distribuée, puis la conclusion.

A.1 État de l'Art

Ce chapitre passe d'un panorama de l'IoT à une analyse ciblée des fondations de la sécurité pour l'IIoT : gestion d'identité et des accès, politiques dynamiques, détection d'anomalies, blockchain industrielle et passeports numériques de produit. L'objectif est d'identifier les exigences spécifiques des environnements industriels (échelle, hétérogénéité, temps réel, intégration d'héritage) et les limites des approches statiques, afin de motiver des solutions adaptatives et distribuées.

IoT vs IIoT, caractéristiques et menaces. L'IIoT étend l'IoT à des domaines critiques (usine, énergie, transport) avec contraintes de latence déterministe, haute disponibilité et sûreté. Les architectures tendent vers l'edge/distribué pour réduire la latence et augmenter la résilience. Les vecteurs d'attaque majeurs incluent ingénierie sociale, malwares (dont ransomware), attaques réseau/protocole (MitM, rejeu, DoS), compromission physique/supply chain et abus d'identifiants.

Identité et authentification. Une identité d'appareil agrège identifiants uniques, attributs descriptifs et clés/certificats $\mathrm{ID}(d) = (U_d, A_d, K_d)$. Son cycle de vie couvre Beginning of Life (BoL), Middle of Life (MoL) et End of Life (EoL) (provisionnement sécurisé, mises à jour/rotation, révocation). Les méthodes d'authentification vont des schémas symétriques et PKI aux approches blockchain et PUF. Les modèles d'Identity Management (IdM) (isolé, centralisé, fédéré, centré utilisateur, décentralisé/Self-Sovereign Identity (SSI) via DIDs/VC) se comparent en scalabilité, sécurité, interopérabilité et effort d'administration.

Contrôle d'accès et politiques. Le contrôle d'accès est formalisé (p. ex. RBAC) et complété par des politiques qui tiennent compte de l'identité, des ressources, des actions et du temps. Les politiques statiques ne suivent ni l'état des dispositifs ni les menaces en évolution. Les politiques dynamiques s'appuient sur des PDPs/PEPs distribués à l'edge et sur la négociation de confiance inter-domaines. Les modèles RBAC, Capability-Based Access Control (CapBAC) et Attribute-Based Access Control (ABAC) sont positionnés pour des usages industriels.

Zero Trust Architecture (ZTA). Le ZTA est un changement de paradigme par rapport à la sécurité périmétrique traditionnelle. Il fonctionne sur le principe de "ne jamais faire confiance, toujours vérifier", ce qui signifie qu'aucun utilisateur ou appareil n'est approuvé par défaut, quel que soit son emplacement sur le réseau. Chaque demande d'accès est authentifiée et autorisée de manière dynamique, en se basant sur des politiques qui tiennent compte de l'identité de l'utilisateur, de l'état de l'appareil, de la localisation et d'autres facteurs contextuels. Les composants clés d'un ZTA incluent le Moteur de Politiques (PE), l'Administrateur de Politiques (PA) et le Point d'Application des Politiques (PEP). Le PE est le cerveau de l'architecture, prenant des décisions d'accès basées sur les politiques définies par le PA. Le PEP est le composant qui applique ces décisions.

Détection d'anomalies réseau. Cinq familles sont passées en revue : à base de connaissances, statistiques, ***machine learning*** supervisé, ***deep learning*** (autoencodeurs, LSTM, hybrides) et graphes/GNNs (y compris graphes temporels et approches contextuelles). Les compromis portent sur faux positifs, adaptation temps réel, empreinte calculatoire et intégration aux politiques.

Blockchain industrielle et variantes légères. La blockchain est un registre distribué, immuable et transparent. Ses principales caractéristiques sont la décentralisation, la transparence, l'immuabilité et la sécurité via la cryptographie. Dans l'IIoT, la blockchain peut être utilisée pour la gestion sécurisée des identités, la traçabilité de la chaîne d'approvisionnement et la création de pistes d'audit inviolables. Les contrats intelligents, des programmes auto-exécutables stockés sur la blockchain, peuvent automatiser l'application des politiques et des accords. Cependant, les implémentations de blockchain traditionnelles comme le Proof-of-Work (PoW) sont gourmandes en ressources, ce qui motive le développement de solutions de blockchain légères pour l'IIoT, telles que celles basées sur le Proof-of-Stake (PoS) ou des structures de graphes acycliques dirigés (DAG).

Passeport Numérique de Produit (DPP). Le DPP agrège les données de cycle de vie pour la circularité (R-stratégies). Couplé à la blockchain, il renforce intégrité et traçabilité, mais pose des enjeux d'interopérabilité, de scalabilité et surtout de protection de la vie privée ; des approches préservent la confidentialité tout en maintenant l'auditabilité.

Synthèse. Les besoins de l'IIoT exigent des capacités intégrées et contextuelles : IAM robuste, politiques dynamiques distribuées, ZTA adaptative, détection d'anomalies temps réel et registres décentralisés légers. Des lacunes persistent (adaptation en ligne, interopérabilité, évaluation continue de la confiance, métriques normalisées), préparant les contributions et cadres proposés dans les chapitres suivants.

A.2 Scénario : Usine Intelligente

Description du Scénario Nous considérons une usine intelligente comme un cas d'utilisation représentatif de l'IIoT. Ce scénario implique plusieurs domaines opérationnels interconnectés, tels que les lignes d'assemblage robotisées, la gestion automatisée des stocks et les outils de maintenance prédictive. Chaque domaine a des exigences de sécurité distinctes mais doit interagir de manière transparente avec les autres pour assurer une production efficace et ininterrompue. Par exemple, les données des capteurs d'une ligne de production peuvent être nécessaires au système de gestion des stocks pour commander automatiquement de nouvelles pièces, tandis que les techniciens de maintenance ont besoin d'un accès à distance aux machines pour le diagnostic.

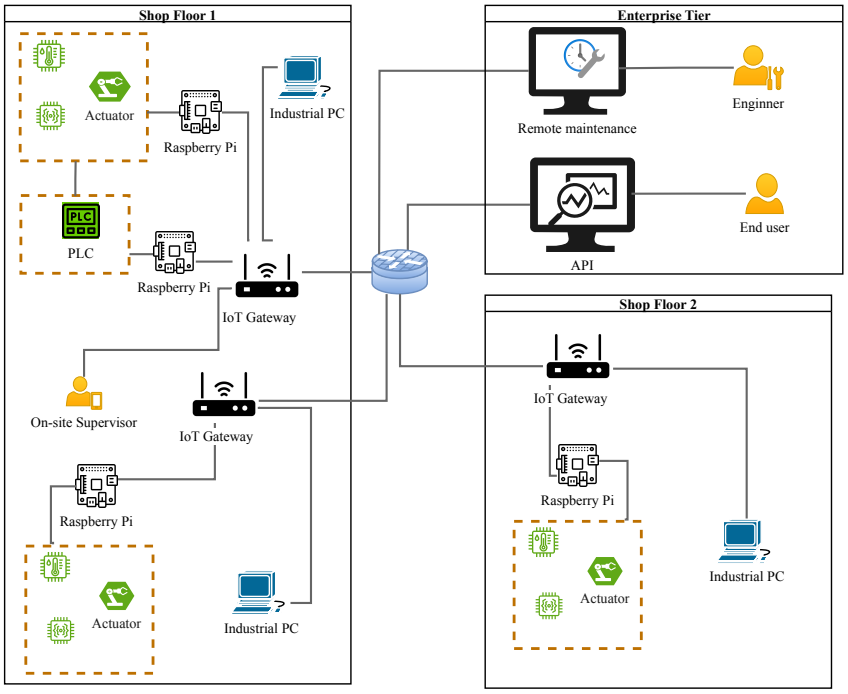


Figure A.1: Cas d'utilisation industriel.

Modèle de Menace Le modèle de menace pour l'usine intelligente (Figure A.2) identifie plusieurs vecteurs d'attaque potentiels :

- **Attaques Externes :** Acteurs non autorisés tentant de pénétrer le réseau de l'usine via Internet, par exemple en exploitant des vulnérabilités dans les services exposés ou par des attaques de phishing ciblant les employés.
- **Menaces Internes :** Employés ou appareils compromis abusant de leurs privilèges d'accès pour saboter les opérations, voler des données sensibles ou introduire des logiciels malveillants dans le réseau.
- **Attaques de la Chaîne d'Approvisionnement :** Compromission de composants ou de logiciels avant leur intégration dans l'usine. Un attaquant pourrait insérer une porte dérobée dans un appareil ou un logiciel fourni par un tiers.
- **Attaques Physiques :** Accès non autorisé à des appareils physiques pour les manipuler, par exemple en connectant un appareil malveillant au réseau local ou en altérant les capteurs.

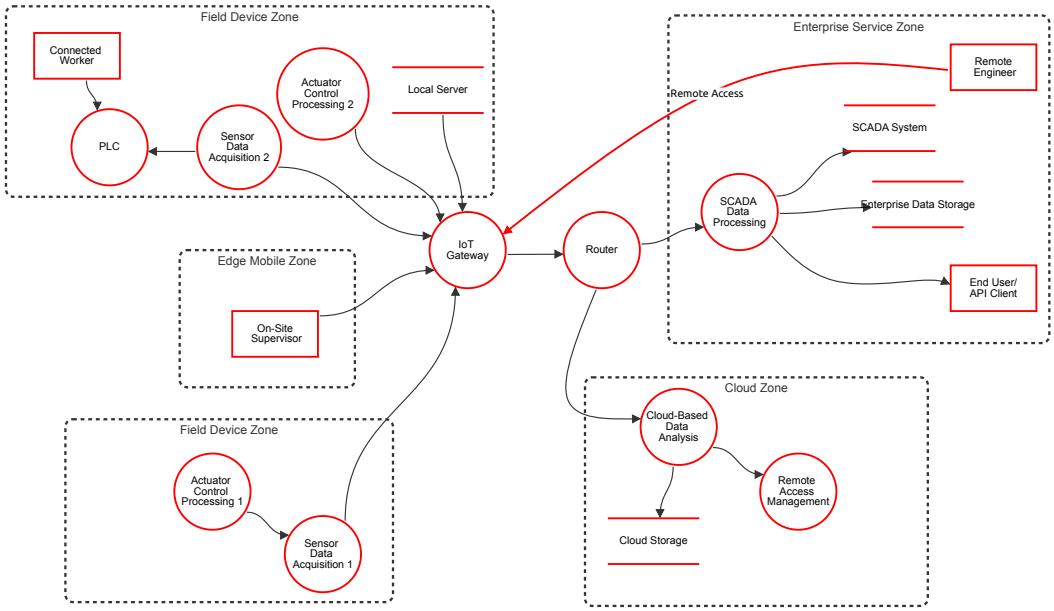


Figure A.2: Modèle de menace pour l'usine intelligente.

A.3 Gestion de l'Identité, Authentification et Politique d'Accès

Gestion Avancée de l'Identité par Portefeuille Numérique pour l'IIoT Cette section présente une architecture de gestion de l'identité basée sur des portefeuilles numériques, conçue pour l'IIoT. Elle classe les appareils en trois catégories de sécurité (faible, modérée, élevée) et gère le cycle de vie des identités à

travers les phases d'enrôlement, d'authentification et de gestion des informations d'identification. Cette approche permet une gestion des identités plus granulaire et adaptée aux risques, tout en étant plus évolutive que les systèmes centralisés traditionnels.

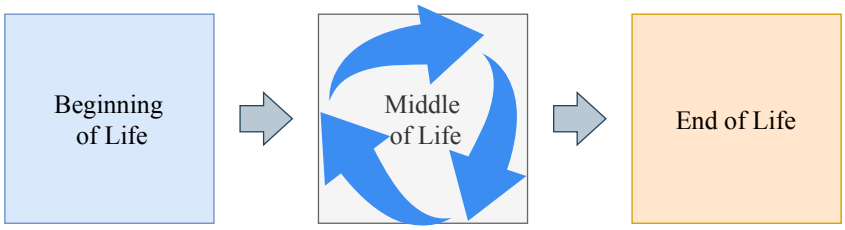


Figure A.3: Cycle de vie de la gestion de l'identité.

Phases de la Gestion de l'Identité Le cycle de vie de la gestion de l'identité dans ce framework est divisé en trois phases principales :

A.3.0.1 Phase 1 : Enrôlement et Émission des Informations d'Identification

Le processus d'enrôlement sécurisé pour les appareils IIoT garantit une authentification forte et un stockage des informations d'identification résistant à la falsification. L'appareil subit une attestation matérielle à l'aide d'une clé d'identité de dispositif vérifiée par TPM, qui est validée par le Nœud Périphérique pour confirmer l'intégrité de l'appareil. En cas de succès, l'appareil s'enregistre auprès du Service de Gestion de l'Identité (IMS), qui attribue un Identifiant Décentralisé (DID) unique et émet des Informations d'Identification Vérifiables (VC). Ces informations sont stockées de manière sécurisée dans le portefeuille numérique de l'appareil, et un enregistrement immuable de l'événement d'enrôlement est écrit sur la blockchain.

A.3.0.2 Phase 2 : Authentification et Contrôle d'Accès

Le processus d'authentification et de contrôle d'accès garantit que seuls les appareils IIoT vérifiés peuvent accéder aux services du réseau. L'appareil initie une demande d'accès en utilisant le TLS mutuel (mTLS) et la signe avec les informations d'identification de son portefeuille numérique. Le Nœud Périphérique valide les preuves d'attestation et les informations d'identification, et interroge le registre de la blockchain pour vérifier le statut d'enregistrement de l'appareil. L'IMS évalue ensuite les informations d'identification par rapport aux politiques de sécurité et au contexte de risque. Si l'évaluation est positive, l'accès est accordé.

A.3.0.3 Phase 3 : Mise à Jour et Révocation des Informations d'Identification

Le processus de mise à jour et de révocation des informations d'identification garantit que le système reste résilient face aux appareils compromis ou se comportant mal.

Les Nœuds Périphériques et l'IMS surveillent en permanence le comportement des appareils. Si une anomalie ou une violation de politique est détectée, le système déclenche une mise à jour ou une révocation des informations d'identification. L'IMS met à jour ou révoque les informations d'identification dans le portefeuille numérique et enregistre l'événement sur le registre de la blockchain. La mise à jour est ensuite propagée à travers les réseaux fédérés pour maintenir la cohérence.

A.4 Authentification Inter-Domaines

Pour relever le défi de la confiance entre les frontières organisationnelles, cette contribution présente un protocole d'authentification amélioré par la blockchain et intégré au TLS. Notre modèle combine les certificats X.509 avec une validation de la confiance gérée par la blockchain. Les nœuds de la blockchain maintiennent des valeurs de confiance dynamiques pour chaque appareil, permettant une authentification rapide et sécurisée même sans validation directe de l'AC. Cette approche est particulièrement utile dans les scénarios où les appareils de différents fabricants ou organisations doivent collaborer en toute sécurité.

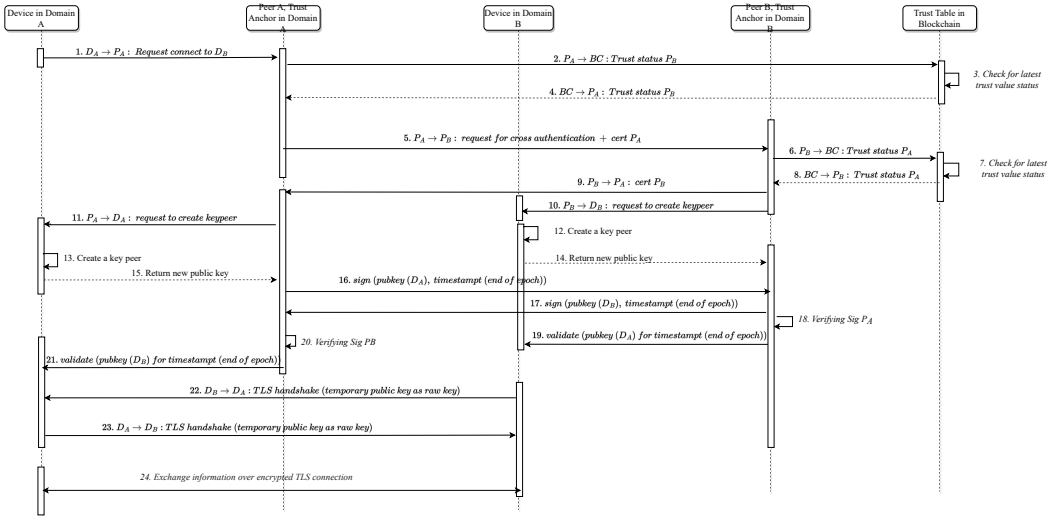


Figure A.4: Diagramme de séquence de l'authentification inter-domaines.

A.5 Détection d'anomalies & évaluation des anomalies dans l'IIoT

Motivation & périmètre Se reposer uniquement sur l'identité et le contrôle d'accès est insuffisant dans des IIoT hétérogènes et contraints en temps. Nous présentons deux détecteurs complémentaires, conçus pour le temps réel et l'échelle : (i) un **modèle hybride AE–LDA** (§5.3) pour détecter les ***zero-day*** avec affectation de classe interprétable ; et (ii) une méthode **sensibles au contexte, basée communautés & multi-graphe** avec un HeteroGNN (§5.4) exploitant le contexte temporel/structurel pour réduire les faux positifs.

Problème en bref Les signatures manquent les menaces nouvelles et génèrent des faux positifs dans des ateliers dynamiques. Les AE seuls surgénéralisent ; l'OC-SVM est fragile en haute dimension ; les AE à mémoire ajoutent de la charge. Besoins : (a) caractéristiques efficaces ; (b) détection robuste + classification ; (c) contexte pour distinguer les changements opérationnels bénins des vraies intrusions.

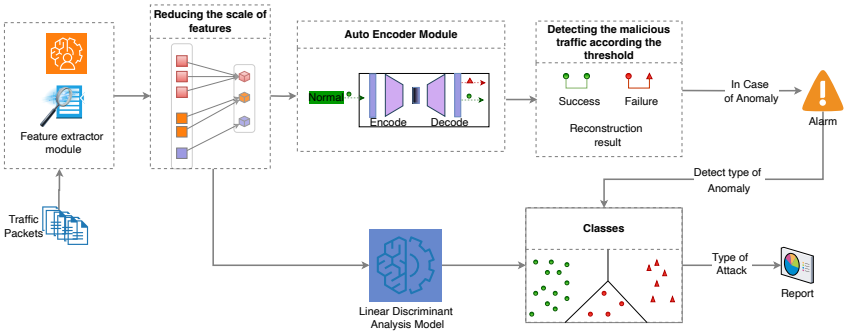


Figure A.5: Le flux de travail du processus de détection d'anomalies.

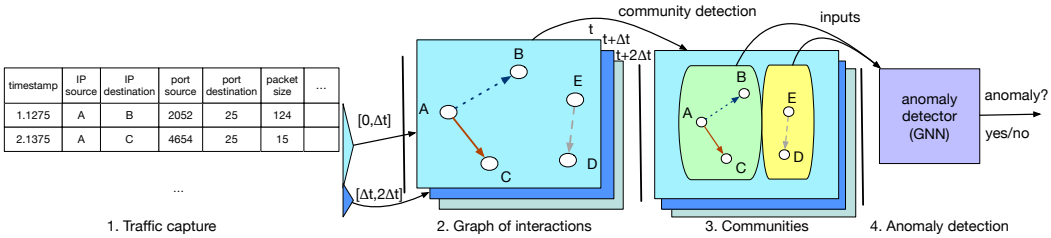


Figure A.6: Étapes de notre approche de détection d'anomalies basée sur la communauté.

Choix de conception clés (très bref) **Caract./explicabilité :** importance Random Forest + SHAP pour des jeux de caractéristiques compacts et transparents. **AE–LDA :** AE entraîné sur flux bénins (seuil MSE) signale l'anomalie ; LDA assigne une classe connue à faible coût. **Contexte :** graphes multi-arêtes évolutifs ; communautés via LPA ; HeteroGNN traite séparément comm/contexte/connaissance ; ***focal loss*** pour le déséquilibre de classes.

A.5.1 Résultats — AE–LDA (anomalies réseau)

Table A.1: CICIDS2017 : AUROC vs bases (plus haut = meilleur).

Modèle	AUROC
OCSVM [91]	0.7684
AE (AE+OCSVM latent) [98]	0.8758
MemAE [94]	0.9101
AE–LDA (nous)	**0.9800**

Table A.2: AE–LDA en ***zero-day*** (entraînement bénin uniquement).

Métrique	Valeur	Remarque
Exactitude	0.9590	entraîné sur bénin seul
F1-score	0.9417	robuste aux attaques nouvelles
Latence (sous-ensemble DoS)	< 12 ms	cf. Table 5.2

Table A.3: Kitsune : AE–LDA vs Griffin (exactitude par scénario).

Scénario	AE–LDA	Griffin [105]
ARP MitM	0.9487	0.8048
Injection Vidéo	0.9007	0.8237
Active Wiretap	0.9669	0.9188
Scan OS	0.9713	0.9281
SSDP Flood	0.9945	**0.9999**

À retenir (AE–LDA). AUROC systématiquement supérieur aux bases sur CICIDS2017 ; compétitif face à Griffin sur Kitsune (meilleur sur 4/5 tâches), avec une inférence en millisecondes adaptée à l'IDS en ligne.

Résultats — HeteroGNN sensible au contexte

Table A.4: Impact des contextes (arêtes) et des communautés.

Configuration	Précision	Rappel	F1	Exact.	AUC
IDS2017 (2 arêtes)	0.8996	1.0000	0.9472	0.9442	0.9973
IDS2017 (3 arêtes)	**0.9972**	**1.0000**	**0.9986**	**0.9986**	0.9973
ToN (2 arêtes)	0.9778	1.0000	0.9888	0.9965	**1.0000**
ToN (3 arêtes)	**0.9888**	**1.0000**	**0.9944**	**0.9982**	**1.0000**
ToN (3 arêtes, sans communautés)	0.9615	0.8523	0.9036	0.9719	0.9922

À retenir (Contexte). Ajouter les arêtes de ***contexte*** et les ***communautés*** augmente nettement la précision/F1 (moins de faux positifs) avec une orchestration sous-seconde et ≈7,5 ms d'inférence : défenses temps réel pour réseaux industriels.

Apports de chaque contribution

- **AE–LDA :** détecte le trafic inédit via AE, puis assigne des catégories connues via LDA pour une triage actionnable ; caractéristiques transparentes (RF+SHAP).
- **HeteroGNN sensible au contexte :** encode ***qui/quand/comment*** communique (multi-arêtes, communautés) pour distinguer maintenance vs intrusion ; améliore la précision sans sacrifier le rappel et conserve le débit temps réel.

Table A.5: Comportement temporel (fenêtre glissante).

Intervalle	Création du graphe	Détection de communautés (LPA)	Mise à jour du graphe	Test GNN
Hebdomadaire	62.14 s	0.052 s	0.504 s	0.0075 s
Horaire	1.87 s	0.0041 s	0.0319 s	0.0075 s
Seconde	0.015 s	0.00002 s	0.00051 s	0.0075 s

Limites & adéquation AE–LDA se concentre sur des caractéristiques réseau (pas de contexte procédé) ; la méthode contexte dépend de la qualité/disponibilité du contexte. Ensemble, elles offrent sensibilité ***zero-day et*** faible taux de faux positifs, alignées avec l'exploitation IIoT.

A.6 Approche blockchain pour sécuriser des environnements industriels distribués

section*Contexte et objectif Les systèmes IIoT mêlent hétérogénéité, distribution et forte exposition aux menaces. Les approches classiques (annuaire central, ACL périmétriques) peinent à fournir ***décentralisation***, ***traçabilité*** et ***inaltérabilité*** sous contraintes d'énergie, calcul et latence. Nous présentons deux contributions blockchain complémentaires et adaptées à ces contraintes : (i) une **blockchain d'atelier (Shopfloor)** qui apporte authentification préservant la vie privée et piste d'audit infalsifiable ; (ii) une **blockchain légère BFT-DAG** offrant haut débit, faible latence et faible coût énergétique pour des nœuds contraints.

Problématique

Les solutions existantes souffrent d'au moins un de ces points : latence/échelle insuffisantes, absence de mécanismes de confidentialité, empreinte calcul/stockage trop élevée, résilience limitée face aux DoS et fautes byzantines. Il faut donc (1) ***déporter*** calcul et stockage loin des nœuds faibles, (2) ***vérifier*** finement des attributs en limitant la divulgation, (3) ***consensuer*** rapidement et sobrement.

Contribution 1 — Shopfloor Blockchain (authentification privée + traçabilité)

L'architecture est ***multi-niveaux*** : **LN** (capteurs/actionneurs) ne gèrent que la capture et l'empreinte des données ; **MN** (nœuds intermédiaires) chiffrent, tamponnent et agrègent via HSMs; **FN** (nœuds complets) assurent le consensus PBFT et l'immutabilité. La ***pré-inscription*** d'un nœud forge une identité forte $ID_n = H(Sn \| MAC)$ (numéro de série + MAC). L'accès et les transactions sont ensuite conditionnés par des ***jetons d'attributs*** (SEC/INS/POC/TRA) délivrés après une preuve n***-sur-***n : le nœud prouve qu'il possède ***tous*** les attributs requis sans les dévoiler inutilement. Les MN bâtissent des blocs candidats signés et ***hors charge LN***; les FN finalisent par PBFT. **Effet clé :** les LN restent ultra-légers

(hachage/empreinte + réponse à défi), tandis que la confidentialité (attributs) et l'auditabilité (chaîne) sont garanties par des rôles plus capacitaires.

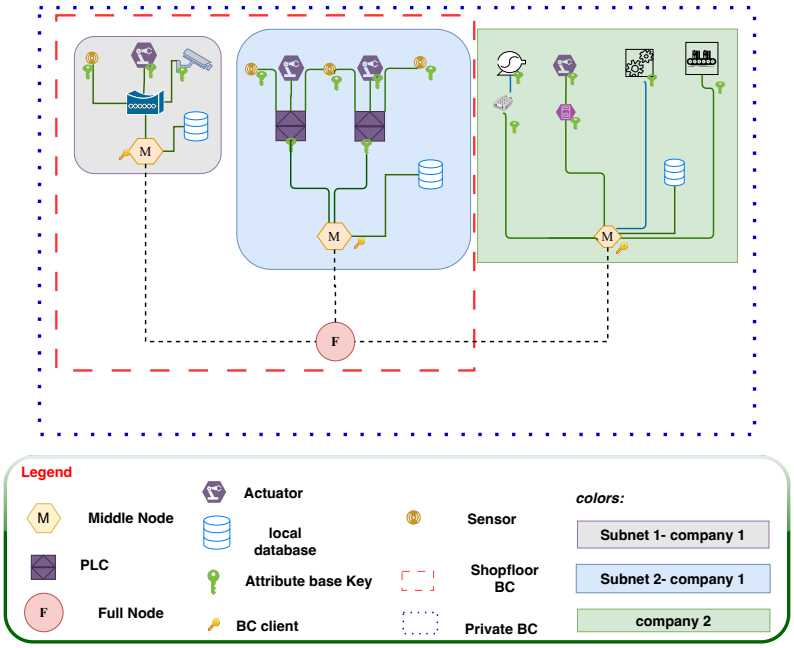


Figure A.7: Architecture Blockchain pour l'atelier de production.

Contribution 2 — Blockchain légère BFT-DAG (débit, latence, sobriété)

Pour les environnements à très forte cadence, nous séparons **RN** (nœuds réguliers, ***prover***), **ECN** (edge, agrégation/filtrage), **TVN** (vérificateurs d'attributs/transactions) et **CN** (comité, BFT-DAG). Le pipeline est le suivant : RN signe → ECN agrège/filtre et pousse en mempool → TVN valide signatures ***et*** jetons d'attributs (sans connaître la donnée) → CN ordonne et engage dans un ***DAG*** par BFT (proposition, pré-commit, commit). L'ordonnancement partiel du DAG autorise une forte parallélisation et des ***finalités rapides***. **Effet clé :** haut débit et faible latence sans faire porter le coût aux RN ; l'***anonymat est volontairement faible*** (traçabilité industrielle), mais la ***secrecy*** d'usage des jetons/clefs est préservée.

Sécurité (synthèse)

Replay : défi TVN avec nonce et timestamp, rendant toute réinjection détectable. **MITM/falsification :** chiffrement bout-à-bout et signatures ; TVN refusent toute transaction sans jeton d'attribut valide. **DoS/DDoS :** pas de point central de contrôle dans la vérification ; tâches réparties (ECN/TVN/CN). **Sybil :** enrôlement par preuves d'attributs et sanctions réputationnelles côté comité. **Canaux auxiliaires :** aléatoirisation et matériel sûr (HSMs) limitent les fuites temporelles/énergétiques.

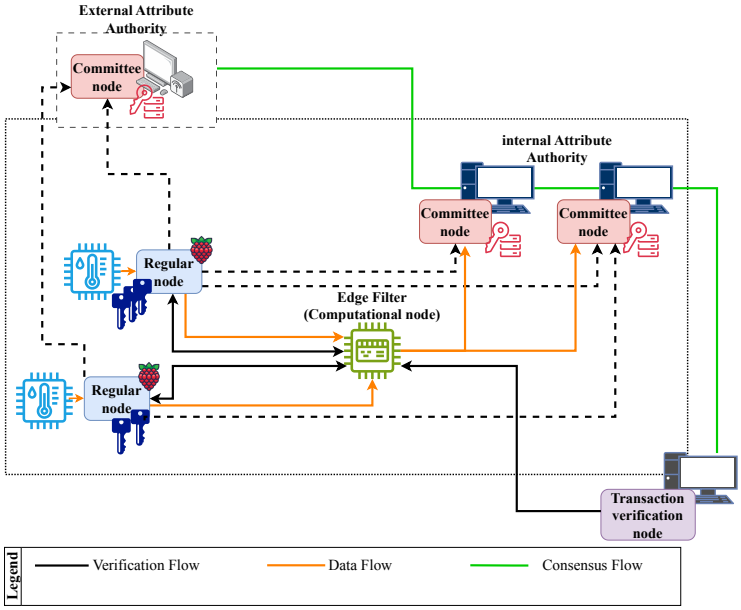


Figure A.8: Relation entre les principaux acteurs de l'architecture proposée.

Évaluation (messages principaux)

Banc Shopfloor : sur un groupe d'essai (7 LN), l'enregistrement reste sub-seconde ; les échanges défi/réponse sont de l'ordre de dizaines de ms et la génération d'empreinte est sous-milliseconde. Le consensus PBFT s'exécute côté FN/MN, sans bloquer l'atelier. **Banc BFT-DAG :** sur un montage hybride (3 laptops comité, 1 ECN, passerelle capteurs), nous atteignons ∼49k TPS avec ∼0.43–0.58 s de latence (consensus vs bout-en-bout). L'énergie par transaction est ≈1.1 mJ, soit un bon compromis ***débit/latence/sobriété*** pour l'IIoT.

Limites et adéquation

Shopfloor suppose des MN/FN dotés d'HSM et de liens fiables ; excellent pour l'audit d'atelier et l'***edge offload***. **BFT-DAG** vise l'efficacité (débit/latence/énergie) avec un anonymat faible assumé pour la conformité et l'investigation. Les deux se combinent : attributs privés, LN minimaux, finalité rapide et registre infalsifiable, ***au rythme de l'atelier***.

A.7 Architecture Zero Trust dynamique (ZTA)

Le chapitre soutient que les modèles Zero Trust statiques, fondés sur des politiques figées, sont mal adaptés aux conditions changeantes de l'IIoT. Il propose une **ZTA dynamique** qui ajuste les décisions d'accès en temps réel en fusionnant des signaux contextuels (posture utilisateur/appareil, contexte de flux, criticité de segment) avec une évaluation continue de la menace.

Table A.6: Synthèse des résultats expérimentaux (plateformes de test).

Approche	Indicateur	Résultat
Shopfloor	Enregistrement (7 LN)	138 ms
Shopfloor	Génération empreinte (1 LN)	0.65 ms
Shopfloor	Validation — défi	38 ms
Shopfloor	Validation — réponse	29 ms
BFT-DAG	Débit consensus	49,439 tx/s (25.31 MB/s)
BFT-DAG	Latence consensus	433 ms
BFT-DAG	Débit bout-en-bout	49,039 tx/s (25.11 MB/s)
BFT-DAG	Latence bout-en-bout	577 ms
BFT-DAG	Énergie/transaction	∼0.00112 J (∼1.1 mJ)

Idée clé

La ZTA dynamique applique « ne jamais faire confiance, toujours vérifier » via :

- **Ajustements dynamiques de confiance** : l'accès est recalculé en continu selon le risque courant.
- **Adaptation continue des politiques** : les règles sont régénérées à partir du contexte et de la menace.
- **Hypothèse d'intégrité de base** : les composants cœur sont supposés sains pour permettre une détection d'anomalies fiable.

Architecture et flux d'accès

Chaque entité (utilisateur, appareil, flux) possède un **ID** unique émis par l'Administration des Politiques (PA). Lors d'une demande d'accès :

1. Le **PEP** vérifie l'ID (liste d'interdiction synchronisée), puis relaie vers l'Authentificateur et le **Policy Engine (PE)**.
2. L'**Authentificateur** valide certificats/identifiants.
3. La **Détection d'anomalies** (réseau et contextuelle) fournit des signaux.
4. Le **Générateur de politiques** met à jour les règles selon l'évaluation de la menace.
5. Le **PE** statue (autoriser/refuser) et le **PEP** applique; tous les événements sont journalisés.

Policy Engine avec machine à états (FSM)

Le PE exécute une **FSM** pour refléter la posture de risque : ***Normal*** → ***Alerte*** → ***Risque élevé*** → ***Quarantaine*** → ***Compromis*** → ***Rétablissement***. Les transitions sont guidées par un **score de Risque de Menace** quantitatif et des événements (anomalie mineure/significative, menace critique, résolution). La FSM rétroalimente les listes d'ID (ban/déban) pour fermer la boucle.

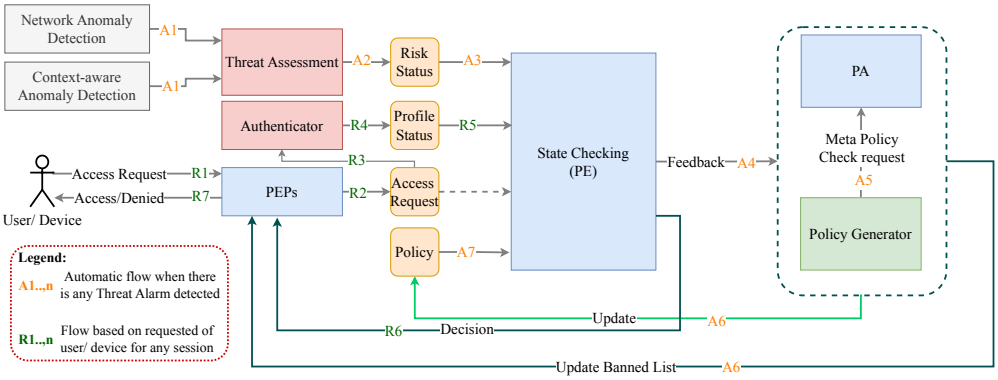


Figure A.9: Framework intégré pour l'architecture Zero Trust dynamique.

Modèle de score de risque

Le risque combine quatre critères normalisés et pondérés w : **Confiance (C)** des détecteurs, **Criticité de l'attaque (A)** (p. ex. cartographie MITRE ATT&CK), **Criticité du segment (S)** (impact métier/sécurité) et **Anomalies passées (P)** (log-transformées et ***robust-scaled***).

$$\text{Risque} = \sum_{v \in \{C,A,S,P\}} w_v \, v_{\text{norm}}$$

Seuils → actions : **Normal** ($< 0{,}4$), **Faible** ($0{,}4$–$0{,}6$), **Élevé** ($0{,}6$–$0{,}8$), **Critique** ($\geq 0{,}8$). Pondérations/seuils initialement définis par experts; futur travail : ajustement data-driven.

Administration des politiques & méta-politiques

La **PA** gère les politiques et des **méta-politiques** (gabarits de haut niveau) et valide les règles via un arbre de décision précompilé, pour cohérence sémantique et ***risk alignment*** (p. ex. `no_write_access_high_risk`). Les conflits sont tranchés par priorité. Complexité maîtrisée : prétraitement $O(|\mathcal{M}|\, a)$, exécution $\approx O(\log |\mathcal{M}| + k)$.

PoC & scénarios

Un PoC sur VMs (PA/PE/PEPs/Authentificateur/Serveur/clients/IoT) montre :

- **Vol d'identifiants** : bloqué par politiques IP/contexte.
- **Escalade interne** : refus par règles rôle+état; anomalie signalée.
- **IoT compromis / DoS** : ***rate limiting*** et élévation d'état isolent l'équipement.
- **Comportement utilisateur suspect** : MFA ou refus déclenché sur déviation.

Évaluation quantitative

L'application des politiques se fait à l'**établissement de session** et aux **points de contrôle**, pas ***par paquet***. Sous charge :

- **Latence** : $\sim 45\,\text{ms}$ (sans ZTA) $\rightarrow \leq 141\,\textbf{ms}$ (avec ZTA), compatible IIoT.
- **CPU** : montée prévisible (pic $\sim 26{,}4\,\%$); **mémoire** ~ 22–$24\,\%$.
- **En DoS** : CPU/mémoire/latence augmentent, le débit baisse; le PE et le serveur restent stables, le PEP ***contient*** la source. Perte de paquets accrue en périphérie.

Déploiement

Pistes : **PEP légers en edge**, **multi-tenant**, **interop** protocoles (MQTT/OPC UA), **scaling horizontal** PEP/PE avec synchronisation distribuée et détection décentralisée.

Conclusion

La ZTA dynamique apporte un Zero Trust **fin, contextuel et réactif** pour l'IIoT, supérieur aux modèles statiques. Les défis restants concernent le coût opérationnel sur dispositifs contraints et le réglage manuel; le chapitre suivant étend l'approche via une **négociation de politiques distribuée et vérifiable par blockchain** pour une sécurité inter-domaine.

A.8 Architecture Zero Trust distribuée (DZTA)

Le chapitre étend les principes Zero Trust vers des environnements *multi-acteurs* et *décentralisés* de l'IIoT, où des organisations autonomes doivent partager données et services sans autorité centrale. Il propose une **DZTA** qui combine une *négociation automatique de politiques* et une **blockchain permissionnée**, en s'appuyant sur des **Digital Product Passports (DPP)** contenant des **Digital Access Policies (DAP)**. L'ancrage sur chaîne apporte traçabilité et auditabilité, tandis que le contenu sensible reste chiffré *off-chain* via des *viewing keys*. La DZTA complète la ZTA dynamique du chapitre précédent en la rendant *inter-organisationnelle*, vérifiable et transparente.

Idée clé et apports

- **Négociation décentralisée de politiques** entre domaines indépendants, avec ancrage immuable des accords.
- **Préservation de la confidentialité** : seules des métadonnées et des références chiffrées sont écrites sur la chaîne; les politiques détaillées restent locales.
- **Interopérabilité et adaptabilité** : décisions alignées sur le contexte, les objectifs métiers et les contraintes réglementaires, sans divulguer la logique interne.

- **Capacités supérieures aux approches existantes** (scalabilité, négociation multi-parties, évaluation quantitative du risque/utilité, *smart contracts*).

Architecture et rôles

Chaque organisation exécute sa propre ZTA (PE/PEP) et participe à une blockchain hiérarchique:

- **Nœuds complets (Full Nodes = PE)** : évaluent les politiques, prennent des décisions et enregistrent l'état consensuel.
- **Nœuds intermédiaires (Middle Nodes = PEP)** : appliquent les décisions, valident l'adhérence et interagissent avec les nœuds complets.
- **Nœuds légers (Light Nodes = entités)** : appareils/utilisateurs conservent localement leurs DPP/DAP et soumettent des informations vers les nœuds intermédiaires.

Les **DPP** encapsulent des métadonnées (configuration, maintenance, cycle de vie) et une couche **DAP** (règles d'accès). Les DPP/DAP sont gérés localement puis *ancrés* sur la chaîne via références chiffrées; seules les parties autorisées peuvent en lire le contenu.

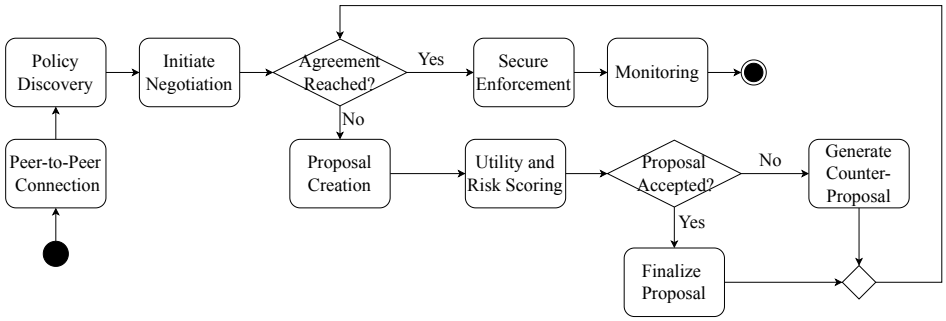


Figure A.10: Flux de données de la méthode proposée.

Garanties de sécurité offertes par la blockchain

- **Intégrité & authenticité** des politiques (signatures, registre immuable).
- **Confidentialité** des contenus (stockage *off-chain* chiffré, contrôle d'accès par clés de visualisation).
- **Non-répudiation & responsabilité** (journalisation signée de toute proposition/acceptation).
- **Disponibilité** de l'historique & de l'état courant (réplication).

Décision, risque et conformité

La négociation s'appuie sur des objectifs quantifiables normalisés dans $[0, 1]$: *sécurité* (contrôles ISO 27001), *réglementaire* (p. ex. RGPD/CCPA), *gouvernance* interne, *vie privée* (minimisation/chiffrement), *confiance* (historique/audits) et *efficacité opérationnelle* (SLA).

Fonction d'utilité.

$$U = \sum_{i=1}^{n} w_i \, f_i(x_i)$$

où w_i reflète l'importance de l'objectif i et $f_i(x_i)$ son niveau de satisfaction normalisé.

Évaluation du risque.

$$R = \frac{\text{Vraisemblance} \times \text{Impact}}{\text{Facteur d'atténuation}}$$

Les propositions à *forte utilité* mais *risque excessif* ($R > \tau_R$) sont rejetées ou soumises à revue humaine.

Conformité.

$$C = \frac{\text{Exigences satisfaites}}{\text{Exigences totales}}$$

L'acceptation exige $U \geq \tau_U$, $R \leq \tau_R$ et $C \geq \tau_C$.

Protocole de négociation (vue d'ensemble)

1. **Initialisation** (mTLS, identité, intégrité du canal).
2. **Récupération/validation** des DAP ancrées (intégrité, signatures, autorisations).
3. **Boucle de négociation** : chaque proposition est notée (U, R, C); si non conforme aux seuils, un *contre-projet* ajustant permissions, preuve d'authentification, durée, quotas, etc. est émis.
4. **Convergence** : mise à jour locale des DAP/PEP et **ancrage on-chain** de l'accord (immutabilité, audit).

Un *smart contract* orchestre états, délais, signatures et finalisation; les objets IIoT restent hors de la boucle blockchain via leurs PEs ou nœuds intermédiaires.

Complexité et scalabilité

Par négociation, le coût dominant vient des **échanges quadratiques** inhérents au consensus BFT; l'évaluation utilité/risque est *légère* (quelques opérations scalaires par proposition). Les empreintes mémoire et trafic restent modestes (politiques *kilo-octets*, quelques tours de négociation), compatibles avec des consortiums d'*ordre de dizaines* d'organisations.

Cas d'usage — chaîne d'approvisionnement

Trois entreprises aux politiques hétérogènes:

- **A** : accès borné par IP et MFA.
- **B** : exposition limitée de paramètres de diagnostic par rôle.
- **C** : données du produit X soumises à anonymisation.

Le flux: requête de collaboration → récupération des DPP/DAP → échanges de priorités/projets → consensus → mise à jour locale (PE/PEP) & ancrage. Résultats: accès d'A aux données de B sous MFA & IP filtrées; B fournit un sous-ensemble de diagnostics; C impose anonymisation. Partage des données chiffré *off-chain*; violations détectées déclenchent alertes ou renégociation.

Analyse de sécurité et hypothèses

Garanties ancrées dans: signatures, mTLS, journalisation immuable, *threshold* BFT ($f < n/3$ malveillants), et hygiène opérationnelle (gestion de clés, disponibilité des nœuds). Limites: attaques hors périmètre sur primitives crypto, compromission de canaux, erreurs de configuration, menaces spécifiques blockchain (p. ex. eclipse) et coûts de performance/storage.

Limites et perspectives

La blockchain introduit *latence* de consensus et *surcoûts* réseau/stockage, sensibles pour des IIoT contraints; la montée en échelle exige un réglage fin. Pistes: optimisation des PEP en périphérie, politiques locatives multi-tenant, interopérabilité protocoles (MQTT/OPC UA), et synchronisation distribuée.

Conclusion

La **DZTA** propose un *cadre fiable, transparent et adaptable* pour la coopération inter-domaine en IIoT, alliant **blockchain** (immutabilité, responsabilité) et **ZTA dynamique** (contexte, réactivité). Malgré des coûts opérationnels liés au consensus, l'approche renforce la confiance, la conformité et la sécurité, et constitue une base crédible pour un déploiement industriel et des travaux futurs sur l'intégration et l'évaluation de performance *in situ*.

A.9 Conclusion et Perspectives de Recherche Future

Récapitulation et Réponse aux Questions de Recherche

Cette thèse a abordé avec succès les principaux défis de la sécurisation des environnements IIoT distribués. En intégrant la gestion de l'identité, la détection d'anomalies, la blockchain et le ZTA, nous avons développé un framework de sécurité complet, adaptatif et évolutif. Les questions de recherche concernant l'intégration de la blockchain et du ZTA, la détection d'anomalies en temps réel, la gestion d'identité évolutive, les limitations du ZTA traditionnel et l'applicabilité de la blockchain dans

l'IIoT ont toutes été traitées à travers les contributions de cette thèse. Le framework intégré qui en résulte offre une défense multicouche qui est à la fois robuste et flexible.

Remarques Finales

Le framework proposé fait progresser l'état de l'art en offrant une stratégie de sécurité cohésive et automatisée pour l'IIoT. Il met l'accent sur la prise de décision adaptative et informée par les risques, s'éloignant des contrôles statiques basés sur le périmètre. Les recherches futures peuvent s'appuyer sur cette base pour affiner davantage les protocoles de négociation de la confiance, optimiser les mécanismes cryptographiques légers et améliorer l'intégration contextuelle en temps réel, ouvrant la voie à des architectures de cybersécurité autonomes et autorégulées pour l'Industrie 4.0.

Bibliography

[1] Fatemeh Stodt and Christoph Reich. "Bridge of Trust: Cross Domain Authentication for Industrial Internet of Things (IIoT) Blockchain over Transport Layer Security (TLS)". In: ***Electronics*** 12.11 (2023), p. 2401.

[2] Fatemeh Stodt and Christoph Reich. "Digital Wallets and Identity Management: Pioneering Advances for Cloud Service Evolution". In: ***International Journal on Advances in Software*** 17.1 (2024), pp. 13–22.

[3] Fatemeh Stodt, Mohammed BM Kamel, Christoph Reich, Fabrice Theoleyre, and Peter Ligeti. "Blockchain-Based Privacy-Preserving Shop Floor Auditing Architecture". In: ***IEEE Access*** 12 (2024), pp. 26747–26758.

[4] Fatemeh Stodt, Mohammed Alshawki, Christoph Reich, Peter Ligeti, and Fabrice Theoleyre. "Securing the Future: Lightweight Blockchain Solutions for IIoT and IoT Networks". In: ***Security and Privacy*** 8.4 (2025), e70070.

[5] Fatemeh Stodt, Christoph Reich, and Fabrice Theoleyre. "Beyond Static Security: A Context-Aware and Real-Time Dynamic Zero Trust Architecture for IIoT Access Control". In: ***IEEE Internet of Things Journal*** (2025).

[6] Fatemeh Stodt and Christoph Reich. "A Review on Digital Wallets and Federated Service for Future of Cloud Services Identity Management". In: ***SERVICE COMPUTATION 2023, The Fifteenth International Conference on Advanced Service Computing, June 26-30, Nice, France***. IARIA. 2023.

[7] Fatemeh Stodt, Fabrice Theoleyre, and Christoph Reich. "Advancing Network Survivability and Reliability: Integrating XAI-Enhanced Autoencoders and LDA for Effective Detection of Unknown Attacks". In: ***2024 20th International Conference on the Design of Reliable Communication Networks (DRCN), May 6-9, Montréal, Canada***. IEEE. 2024, pp. 9–16.

[8] Fatemeh Stodt, Philipp Ruf, and Christoph Reich. "Blockchain-Enabled Digital Product Passports for Enhancing Security and Lifecycle Management in Healthcare Devices". In: ***2024 8th Cyber Security in Networking Conference (CSNet)***. IEEE. 2024, pp. 44–51.

[9] Cristina Alcaraz, Javier Lopez, Jianying Zhou, and Rodrigo Roman. "Secure SCADA framework for the protection of energy control systems". In: ***Concurrency and Computation: Practice and Experience*** 23.12 (2011), pp. 1431–1442.

[10] Manuel Cheminod, Luca Durante, and Adriano Valenzano. "Review of security issues in industrial networks". In: ***IEEE transactions on industrial informatics*** 9.1 (2012), pp. 277–293.

[11] Ralph Langner. "Stuxnet: Dissecting a cyberwarfare weapon". In: ***IEEE security & privacy*** 9.3 (2011), pp. 49–51.

[12] Alessandro Di Pinto, Younes Dragoni, and Andrea Carcano. "TRITON: The first ICS cyber attack on safety instrument systems". In: ***Proc. Black Hat USA*** 2018 (2018), pp. 1–26.

[13] Thomas Miller, Alexander Staves, Sam Maesschalck, Miriam Sturdee, and Benjamin Green. "Looking back to look forward: Lessons learnt from cyber-attacks on industrial control systems". In: ***International Journal of Critical Infrastructure Protection*** 35 (2021), p. 100464.

[14] Giancarlo Fortino, Antonio Guerrieri, Wilma Russo, and Claudio Savaglio. "Integration of agent-based and cloud computing for the smart objects-oriented IoT". In: ***Proceedings of the 2014 IEEE 18th international conference on computer supported cooperative work in design (CSCWD)***. IEEE. 2014, pp. 493–498.

[15] Mahmoud Ammar, Giovanni Russello, and Bruno Crispo. "Internet of Things: A survey on the security of IoT frameworks". In: ***Journal of information security and Applications*** 38 (2018), pp. 8–27.

[16] Vishal A Thakor, Mohammad Abdur Razzaque, and Muhammad RA Khandaker. "Lightweight cryptography algorithms for resource-constrained IoT devices: A review, comparison and research opportunities". In: ***IEEE Access*** 9 (2021), pp. 28177–28193.

[17] Yuchen Yang, Longfei Wu, Guisheng Yin, Lijie Li, and Hongbin Zhao. "A survey on security and privacy issues in Internet-of-Things". In: ***IEEE Internet of things Journal*** 4.5 (2017), pp. 1250–1258.

[18] V Stafford. "Zero trust architecture". In: ***NIST special publication*** 800 (2020), p. 207.

[19] Allison Wylde. "Zero trust: Never trust, always verify". In: ***2021 international conference on cyber situational awareness, data analytics and assessment (cybersa)***. IEEE. 2021, pp. 1–4.

[20] Konstantinos Christidis and Michael Devetsikiotis. "Blockchains and smart contracts for the internet of things". In: ***IEEE access*** 4 (2016), pp. 2292–2303.

[21] Sachchidanand Singh and Nirmala Singh. "Blockchain: Future of financial and cyber security". In: ***2016 2nd international conference on contemporary computing and informatics (IC3I)***. IEEE. 2016, pp. 463–467.

[22] Luigi Atzori, Antonio Iera, and Giacomo Morabito. "The internet of things: A survey". In: ***Computer networks*** 54.15 (2010), pp. 2787–2805.

[23] Hugh Boyes, Bil Hallaq, Joe Cunningham, and Tim Watson. "The industrial internet of things (IIoT): An analysis framework". In: ***Computers in industry*** 101 (2018), pp. 1–12.

[24] Xingjie Yu and Huaqun Guo. "A survey on IIoT security". In: ***2019 IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS)***. IEEE. 2019, pp. 1–5.

[25] Akseer Ali Mirani, Gustavo Velasco-Hernandez, Anshul Awasthi, and Joseph Walsh. "Key challenges and emerging technologies in industrial IoT architectures: A review". In: ***Sensors*** 22.15 (2022), p. 5836.

[26] Björn Leander, Aida Čaušević, and Hans Hansson. "Applicability of the IEC 62443 standard in Industry 4.0/IIoT". In: ***Proceedings of the 14th International Conference on Availability, Reliability and Security***. 2019, pp. 1–8.

[27] Alparslan Sari, Alexios Lekidis, and Ismail Butun. "Industrial networks and IIoT: Now and future trends". In: ***Industrial IoT: Challenges, Design Principles, Applications, and Security*** (2020), pp. 3–55.

[28] Muhammad Muzamil Aslam, Kassim Kalinaki, Ali Tufail, Abdul Ghani Haji Naim, Madiha Zahir Khan, and Sajid Ali. "Social Engineering Attacks in Industrial Internet of Things and Smart Industry: Detection and Prevention". In: ***Emerging Threats and Countermeasures in Cybersecurity*** (2025), pp. 389–412.

[29] Yash Shah and Shamik Sengupta. "A survey on Classification of Cyberattacks on IoT and IIoT devices". In: ***2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)***. IEEE. 2020, pp. 0406–0413.

[30] Konstantinos Tsiknas, Dimitrios Taketzis, Konstantinos Demertzis, and Charalabos Skianis. "Cyber threats to industrial IoT: a survey on attacks and countermeasures". In: ***IoT*** 2.1 (2021), pp. 163–186.

[31] José Cecílio and André Souto. "Security issues in industrial Internet-of-Things: Threats, attacks and solutions". In: ***2024 IEEE International Workshop on Metrology for Industry 4.0 & IoT (MetroInd4. 0 & IoT)***. IEEE. 2024, pp. 458–463.

[32] SC Vetrivel, R Maheswari, and TP Saravanan. "Industrial IOT: Security Threats and Counter Measures". In: ***Communication Technologies and Security Challenges in IoT: Present and Future***. Springer, 2024, pp. 403–425.

[33] Sana Abdelaziz Bkheet and Johnson I Agbinya. "A review of identity methods of internet of things (IOT)". In: ***Advances in Internet of Things*** 11.4 (2021), pp. 153–174.

[34] Daniel Del Gaudio, Maximilian Reichel, and Pascal Hirmer. "A Life Cycle Method for Device Management in Dynamic IoT Environments." In: ***IoTBDS***. 2020, pp. 46–56.

[35] Tarak Nandy, Mohd Yamani Idna Bin Idris, Rafidah Md Noor, Laiha Mat Kiah, Lau Sian Lun, Nor Badrul Annuar Juma'at, Ismail Ahmedy, Norjihan Abdul Ghani, and Sananda Bhattacharyya. "Review on security of internet of things authentication mechanism". In: ***IEEE Access*** 7 (2019), pp. 151054–151089.

[36] Dongyang Xu, Keping Yu, and James A Ritcey. "Cross-layer device authentication with quantum encryption for 5G enabled IIoT in industry 4.0". In: ***IEEE Transactions on Industrial Informatics*** 18.9 (2021), pp. 6368–6378.

[37] Meng Shen, Huisen Liu, Liehuang Zhu, Ke Xu, Hongbo Yu, Xiaojiang Du, and Mohsen Guizani. "Blockchain-assisted secure device authentication for cross-domain industrial IoT". In: ***IEEE Journal on Selected Areas in Communications*** 38.5 (2020), pp. 942–954.

[38] Shipeng Gao, Yuemin Ding, Yaqi Lu, Li Han, Lei Zhou, Chao Chen, Xiaohan Yu, and Xuefei Huang. "A lightweight fingerprint-based device authentication architecture for wireless industrial automation networks". In: ***2019 1st International Conference on Industrial Artificial Intelligence (IAI)***. IEEE. 2019, pp. 1–6.

[39] Md Sadek Ferdous, Gethin Norman, and Ron Poet. "Mathematical modelling of identity, identity management and other related topics". In: ***Proceedings of the 7th International Conference on Security of Information and Networks***. 2014, pp. 9–16.

[40] Daniela Pöhn and Wolfgang Hommel. "IMC: A Classification of Identity Management Approaches". In: ***arXiv preprint arXiv:2301.00444*** (2023).

[41] Mario Montagud, Fernando Boronat, Hans Stokking, and Pablo Cesar. "Design, development and assessment of control schemes for IDMS in a standardized RTCP-based solution". In: ***Computer Networks*** 70 (2014), pp. 240–259.

[42] Roberto Baldoni. "Federated Identity Management systems in e-government: the case of Italy". In: ***Electronic Government, an International Journal*** 9.1 (2012), pp. 64–84.

[43] Bart Priem, Eleni Kosta, Aleksandra Kuczerawy, Jos Dumortier, and Ronald Leenes. "User-centric privacy-enhancing identity management". In: ***Digital Privacy: PRIME-Privacy and Identity Management for Europe*** (2011), pp. 91–106.

[44] Bruno Cremonezi, Alex B Vieira, José Nacif, Edelberto Franco Silva, and Michele Nogueira. "Identity management for Internet of Things: Concepts, challenges and opportunities". In: ***Computer Communications*** (2024).

[45] Joint Task Force. ***Security and privacy controls for information systems and organizations***. Tech. rep. National Institute of Standards and Technology, 2017.

[46] Antonios Gouglidis, Christos Grompanopoulos, and Anastasia Mavridou. "Formal verification of usage control models: a case study of UseCON using TLA+". In: ***arXiv preprint arXiv:1806.09848*** (2018).

[47] Michael Howlett and Michael Ramesh. "Designing for adaptation: Static and dynamic robustness in policy-making". In: ***Public Administration*** 101.1 (2023), pp. 23–35.

[48] Mersedeh Sadeghi, Luca Sartor, and Matteo Rossi. "A semantic-based access control mechanism for distributed systems". In: ***Proceedings of the 36th Annual ACM Symposium on Applied Computing***. 2021, pp. 1864–1873.

[49] Fan Li, Gary White, and Siobhán Clarke. "A trust model for SLA negotiation candidates selection in a dynamic IoT environment". In: ***IEEE Transactions on Services Computing*** 15.5 (2021), pp. 2565–2578.

[50] Kent E Seamons, Marianne Winslett, Ting Yu, Bryan Smith, Evan Child, Jared Jacobson, Hyrum Mills, and Lina Yu. "Requirements for policy languages for trust negotiation". In: ***Proceedings Third International Workshop on Policies for Distributed Systems and Networks***. IEEE. 2002, pp. 68–79.

[51] Dorota Filipczuk, Tim Baarslag, Enrico H Gerding, and MC Schraefel. "Automated privacy negotiations with preference uncertainty". In: ***Autonomous Agents and Multi-Agent Systems*** 36.2 (2022), p. 49.

[52] Kallol Krishna Karmakar, Vijay Varadharajan, and Uday Tupakula. "Policy-Driven Security Architecture for Internet of Things (IoT) Infrastructure". In: ***Internet of Things Security and Privacy***. CRC Press, 2023, pp. 76–120.

[53] Xiaoyan Hu, Wenjie Gao, Guang Cheng, Ruidong Li, Yuyang Zhou, and Hua Wu. "Towards early and accurate network intrusion detection using graph embedding". In: ***IEEE Transactions on Information Forensics and Security*** (2023).

[54] Gunjan Batra. "Attribute-Based Access Control". In: ***Encyclopedia of Cryptography, Security and Privacy***. Springer, 2024, pp. 1–3.

[55] Antonios Gouglidis and Ioannis Mavridis. "domRBAC: An access control model for modern collaborative systems". In: ***computers & security*** 31.4 (2012), pp. 540–556.

[56] Parikshit N Mahalle, Bayu Anggorojati, Neeli R Prasad, and Ramjee Prasad. "Identity authentication and capability based access control (iacac) for the internet of things". In: ***Journal of Cyber Security and Mobility*** 1.4 (2013), pp. 309–348.

[57] John Kindervag et al. "Build security into your network's dna: The zero trust network architecture". In: ***Forrester Research Inc*** 27 (2010), pp. 1–16.

[58] Scott Rose, Oliver Borchert, Stu Mitchell, and Sean Connelly. "NIST special publication 800-207 zero trust architecture". In: ***NIST National Institute of Standards and Technology US Department of Commerce*** (2020), pp. 800–207.

[59] Evan Gilman. ***Zero trust networks: Building systems in untrusted networks***. O'Reilly, 2016.

[60] Xiangshuai Yan and Huijuan Wang. "Survey on zero-trust network security". In: ***International Conference Artificial Intelligence and Security (ICAIS)***. Hohhot, China, July 2020, pp. 50–60.

[61] Elisa Bertino. "Zero trust architecture: does it help?" In: ***IEEE Security & Privacy*** 19.05 (2021), pp. 95–96.

[62] Barclay Osborn, Justin McWilliams, Betsy Beyer, and Max Saltonstall. "BeyondCorp: Design to Deployment at Google". In: ***;login: (USENIX)*** 41.1 (2016), pp. 28–34. URL: https://www.usenix.org/publications/login/spring2016/osborn.

[63] Eduardo B Fernandez and Andrei Brazhuk. "A critical analysis of Zero Trust Architecture (ZTA)". In: ***Computer Standards & Interfaces*** 89 (2024), p. 103832.

[64] Leonard Bradatsch, Oleksandr Miroshkin, and Frank Kargl. "ZTSFC: A Service Function Chaining-Enabled Zero Trust Architecture". In: ***IEEE Access*** (2023).

[65] Sungmin Hong, Lei Xu, Jianwei Huang, Hongda Li, Hongxin Hu, and Guofei Gu. "SysFlow: Toward a programmable zero trust framework for system security". In: ***IEEE Transactions on Information Forensics and Security*** 18 (2023), pp. 2794–2809.

[66] Zhi Niu, Luming Dong, and Yong Zhu. "The Runtime model checking Method for Zero Trust Security Policy". In: ***International Conference on Cyber Security and Information Engineering (ICCSIE)***. 2022, pp. 8–12.

[67] TLA+ Community. ***TLA+ GitHub Repository***. Accessed: 2025-04-01. 2025. URL: https://github.com/tlaplus.

[68] Qigui Yao, Qi Wang, Xiaojian Zhang, and Jiaxuan Fei. "Dynamic access control and authorization system based on zero-trust architecture". In: ***Proceedings of the 2020 1st international conference on control, robotics and intelligent system***. 2020, pp. 123–127.

[69] Giovanni R da Silva, Daniel F Macedo, and Aldri L dos Santos. "Zero trust access control with context-aware and behavior-based continuous authentication for smart homes". In: ***SBC***. 2021, pp. 43–56. DOI: 10.5753/sbseg.2021.17305. URL: https://doi.org/10.5753/sbseg.2021.17305.

[70] Zirak Zaheer, Hyunseok Chang, Sarit Mukherjee, and Jacobus Van der Merwe. "eztrust: Network-independent zero-trust perimeterization for microservices". In: ***Symposium on SDN Research (SOSR)***. ACM. 2019, pp. 49–61.

[71] Xiaomeng Feng and Shiyan Hu. "Cyber-physical zero trust architecture for industrial cyber-physical systems". In: ***IEEE Transactions on Industrial Cyber-Physical Systems*** 1 (2023), pp. 394–405.

[72] Claudio Zanasi, Silvio Russo, and Michele Colajanni. "Flexible zero trust architecture for the cybersecurity of industrial IoT infrastructures". In: ***Ad Hoc Networks*** 156 (2024), p. 103414.

[73] Fabio Federici, Davide Martintoni, and Valerio Senni. "A zero-trust architecture for remote access in industrial IoT infrastructures". In: ***Electronics*** 12.3 (2023), p. 566.

[74] Biplob Paul and Muzaffar Rao. "Zero-trust model for smart manufacturing industry". In: ***Applied Sciences*** 13.1 (2022), p. 221.

[75] Shiyu Xiao, Yuhang Ye, Nadia Kanwal, Thomas Newe, and Brian Lee. "SoK: context and risk aware access control for zero trust systems". In: ***Security and Communication Networks*** 2022.1 (2022), p. 7026779.

[76] Naeem Firdous Syed, Syed W Shah, Arash Shaghaghi, Adnan Anwar, Zubair Baig, and Robin Doss. "Zero trust architecture (zta): A comprehensive survey". In: ***IEEE access*** 10 (2022), pp. 57143–57179.

[77] Meha James, Thomas Newe, Donna O'Shea, and George D O'Mahony. "Authentication and Authorization in Zero Trust IoT: A Survey". In: ***2024 35th Irish Signals and Systems Conference (ISSC)***. IEEE. 2024, pp. 1–7.

[78] Aman Kumar Routh and Prabhat Ranjan. "A Comprehensive Review on Granularity Perspective of the Access Control Models in Cloud Computing". In: ***2024 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI)***. Vol. 2. IEEE. 2024, pp. 1–6.

[79] Nastaran Farhadighalati, Luis A Estrada-Jimenez, Sanaz Nikghadam-Hojjati, and Jose Barata. "A Systematic Review of Access Control Models: Background, Existing Research, and Challenges". In: ***IEEE Access*** (2025).

[80] Youngho Kim, Seon-Gyoung Sohn, Hae Sook Jeon, Sang-Min Lee, Yunkyung Lee, and Jeongnyeo Kim. "Exploring Effective Zero Trust Architecture for Defense Cybersecurity: A Study". In: ***KSII Transactions on Internet and Information Systems (TIIS)*** 18.9 (2024), pp. 2665–2691.

[81] Steven J Templeton and Karl E Levitt. "Detecting spoofed packets". In: ***Proceedings DARPA Information Survivability Conference and Exposition***. Vol. 1. IEEE. 2003, pp. 164–175.

[82] Ri Wang, Chen Li, Kun Zhang, and Bibo Tu. "Zero-trust based dynamic access control for cloud computing". In: ***Cybersecurity*** 8.1 (2025), p. 12.

[83] Shay Reardon, Murtadha D Hssayeni, and Imadeldin Mahgoub. "Detection of zero-day attacks on iot". In: ***2024 International Conference on Smart Applications, Communications and Networking (SmartNets)***. IEEE. 2024, pp. 1–5.

[84] Thavavel Vaiyapuri, Zohra Sbai, Haya Alaskar, and Nourah Ali Alaseem. "Deep learning approaches for intrusion detection in IIoT networks–opportunities and future directions". In: ***International Journal of Advanced Computer Science and Applications*** 12.4 (2021).

[85] Nicholas Jeffrey, Qing Tan, and José R Villar. "A review of anomaly detection strategies to detect threats to cyber-physical systems". In: ***Electronics*** 12.15 (2023), p. 3283.

[86] Yunyun Hou, Ruiyu He, Jie Dong, Yangrui Yang, and Wei Ma. "Iot anomaly detection based on autoencoder and bayesian gaussian mixture model". In: ***Electronics*** 11.20 (2022), p. 3287.

[87] Fangyu Li, Aditya Shinde, Yang Shi, Jin Ye, Xiang-Yang Li, and Wenzhan Song. "System statistics learning-based IoT security: Feasibility and suitability". In: ***IEEE Internet of Things Journal*** 6.4 (2019), pp. 6396–6403.

[88] Mahsa Raeiszadeh, Amin Ebrahimzadeh, Roch H Glitho, Johan Eker, and Raquel AF Mini. "Real-Time Adaptive Anomaly Detection in Industrial IoT Environments". In: ***IEEE Transactions on Network and Service Management*** (2024).

[89] Weijie Hao, Tao Yang, and Qiang Yang. "Hybrid statistical-machine learning for real-time anomaly detection in industrial cyber–physical systems". In: ***IEEE Transactions on Automation Science and Engineering*** 20.1 (2021), pp. 32–46.

[90] Aliaa Al-Bakaa and Bahaa Al-Musawi. "A new intrusion detection system based on using non-linear statistical analysis and features selection techniques". In: ***Computers & Security*** 122 (2022), p. 102906.

[91] Bernhard Schölkopf, Robert C Williamson, Alex Smola, John Shawe-Taylor, and John Platt. "Support vector method for novelty detection". In: ***Advances in neural information processing systems*** 12 (1999).

[92] Igor Fosić, Drago Žagar, Krešimir Grgić, and Višnja Križanović. "Anomaly detection in NetFlow network traffic using supervised machine learning algorithms". In: ***Journal of Industrial Information Integration*** 33 (2023), p. 100466. ISSN: 2452-414X. DOI: 10.1016/j.jii.2023.100466.

[93] Sultan Zavrak and Murat Iskefiyeli. "Anomaly-based intrusion detection from network flow features using variational autoencoder". In: ***IEEE Access*** 8 (2020), pp. 108346–108358.

[94] Byeongjun Min, Jihoon Yoo, Sangsoo Kim, Dongil Shin, and Dongkyoo Shin. "Network anomaly detection using memory-augmented deep autoencoder". In: ***IEEE Access*** 9 (2021), pp. 104695–104706.

[95] Imtiaz Ullah and Qusay H. Mahmoud. "Design and Development of RNN Anomaly Detection Model for IoT Networks". In: ***IEEE Access*** 10 (2022), pp. 62722–62750. DOI: 10.1109/ACCESS.2022.3176317.

[96] Hoang Duy Trinh, Lorenza Giupponi, and Paolo Dini. "Urban Anomaly Detection by processing Mobile Traffic Traces with LSTM Neural Networks". In: ***2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)***. 2019, pp. 1–8. DOI: 10.1109/SAHCN.2019.8824981.

[97] Mahmoud Said Elsayed, Nhien-An Le-Khac, Soumyabrata Dev, and Anca Delia Jurcut. "Network Anomaly Detection Using LSTM Based Autoencoder". In: ***Symposium on QoS and Security for Wireless and Mobile Networks (Q2SWinet)***. ACM. 2020, pp. 37–45. DOI: 10.1145/3416013.3426457.

[98] Lotfi Mhamdi, Desmond McLernon, Fadi El-Moussa, Syed Ali Raza Zaidi, Mounir Ghogho, and Tuan Tang. "A deep learning approach combining autoencoder with one-class SVM for DDoS attack detection in SDNs". In: ***2020 IEEE Eighth International Conference on Communications and Networking (ComNet)***. IEEE. 2020, pp. 1–6.

[99] Amiya Kumar Sahu, Suraj Sharma, M. Tanveer, and Rohit Raja. "Internet of Things attack detection using hybrid Deep Learning Model". In: ***Computer Communications*** 176 (2021), pp. 146–154. ISSN: 0140-3664. DOI: 10.1016/j.comcom.2021.05.024.

[100] Chenyang Qiu, Yingsheng Geng, Junrui Lu, Kaida Chen, Shitong Zhu, Ya Su, Guoshun Nan, Can Zhang, Junsong Fu, Qimei Cui, et al. "3D-IDS: Doubly Disentangled Dynamic Intrusion Detection". In: ***ACM SIGKDD***. 2023, pp. 1965–1977.

[101] Roberto Doriguzzi-Corin, Stuart Millar, Sandra Scott-Hayward, Jesus Martinez-del Rincon, and Domenico Siracusa. "LUCID: A practical, lightweight deep learning solution for DDoS attack detection". In: ***IEEE Transactions on Network and Service Management*** 17.2 (2020), pp. 876–889.

[102] Chong Zhou and Randy C Paffenroth. "Anomaly detection with robust deep autoencoders". In: ***international conference on knowledge discovery and data mining***. ACM SIGKDD. 2017, pp. 665–674.

[103] Dongqi Wang, Mingshuo Nie, and Dongming Chen. "BAE: Anomaly Detection Algorithm Based on Clustering and Autoencoder". In: ***Mathematics*** 11.15 (2023), p. 3398.

[104] Yisroel Mirsky, Tomer Doitshman, Yuval Elovici, and Asaf Shabtai. "Kitsune: an ensemble of autoencoders for online network intrusion detection". In: ***arXiv preprint arXiv:1802.09089*** https://doi.org/10.24432/C5D90Q (2018).

[105] Liyan Yang, Yubo Song, Shang Gao, Aiqun Hu, and Bin Xiao. "Griffin: Real-time network intrusion detection system via ensemble of autoencoder in SDN". In: ***IEEE Transactions on Network and Service Management*** 19.3 (2022), pp. 2269–2281.

[106] Claude Berge. ***The Theory of Graphs***. Wiley, 1962.

[107] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. "A Comprehensive Survey on Graph Neural Networks". In: ***IEEE Transactions on Neural Networks and Learning Systems*** 32.1 (2021), pp. 4–24. DOI: 10.1109/TNNLS.2020.2978386.

[108] Xiaokang Zhou, Wei Liang, Weimin Li, Ke Yan, Shohei Shimizu, and Kevin I-Kai Wang. "Hierarchical Adversarial Attacks Against Graph-Neural-Network-Based IoT Network Intrusion Detection System". In: ***IEEE Internet of Things Journal*** 9.12 (2022), pp. 9310–9319. DOI: 10.1109/JIOT.2021.3130434.

[109] Yulei Wu, Hong-Ning Dai, and Haina Tang. "Graph neural networks for anomaly detection in industrial internet of things". In: ***IEEE Internet of Things Journal*** 9.12 (2021), pp. 9214–9231.

[110] Laetitia Leichtnam, Eric Totel, Nicolas Prigent, and Ludovic Mé. "Sec2graph: Network attack detection based on novelty detection on graph structured data". In: ***International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA)***. Springer. Lisbon, Portugal, 2020, pp. 238–258.

[111] Wai Weng Lo, Siamak Layeghy, Mohanad Sarhan, Marcus Gallagher, and Marius Portmann. "E-graphsage: A graph neural network based intrusion detection system for iot". In: ***NOMS***. IEEE/IFIP. 2022, pp. 1–9. DOI: 10.1109/NOMS54207.2022.9789878.

[112] Lieqing Lin, Qi Zhong, Jiasheng Qiu, and Zhenyu Liang. "E-GRACL: an IoT intrusion detection system based on graph neural networks". In: ***The Journal of Supercomputing*** 81.1 (2025), p. 42.

[113] Tanzeela Altaf, Xu Wang, Wei Ni, Guangsheng Yu, Ren Ping Liu, and Robin Braun. "A new concatenated Multigraph Neural Network for IoT intrusion detection". In: ***Internet of Things*** 22 (2023), p. 100818. DOI: 10.1016/j.iot.2023.100818.

[114] Hamdi Friji, Alexis Olivereau, and Mireille Sarkiss. "Efficient Network Representation for GNN-Based Intrusion Detection". In: ***ACNS***. Springer. 2023, pp. 532–554.

[115] Chanyoung Park, Donghyun Kim, Jiawei Han, and Hwanjo Yu. "Unsupervised attributed multiplex network embedding". In: ***AAAI***. Vol. 34. 2020, pp. 5371–5378.

[116] Minji Yoon, Bryan Hooi, Kijung Shin, and Christos Faloutsos. "Fast and accurate anomaly detection in dynamic graphs with a two-pronged approach". In: ***ACM SIGKDD***. 2019, pp. 647–657.

[117] Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael Bronstein. "Temporal graph networks for deep learning on dynamic graphs". In: ***ICML Workshop on Graph Representation Learning***. 2020, pp. 1–9.

[118] Isaiah J King and H Howie Huang. "Euler: Detecting network lateral movement via scalable temporal link prediction". In: ***ACM Transactions on Privacy and Security*** 26.3 (2023), pp. 1–36.

[119] Anasua Mitra, Priyesh Vijayan, Ranbir Sanasam, Diganta Goswami, Srinivasan Parthasarathy, and Balaraman Ravindran. "Semi-supervised deep learning for multiplex networks". In: ***ACM SIGKDD***. 2021, pp. 1234–1244.

[120] Paweł Kowalski and Anne-Laure Jousselme. "Context-awareness for information correction and reasoning in evidence theory". In: ***International Journal of Approximate Reasoning*** 153 (2023), pp. 29–48.

[121] Antonino Rullo, Daniele Midi, Anand Mudjerikar, and Elisa Bertino. "Kalis2.0—A SECaaS-Based Context-Aware Self-Adaptive Intrusion Detection System for IoT". In: ***IEEE Internet of Things Journal*** 11.7 (2024), pp. 12579–12601. DOI: 10.1109/JIOT.2023.3333948.

[122] Rozhin Yasaei, Felix Hernandez, and Mohammad Abdullah Al Faruque. "IoT-CAD: Context-aware adaptive anomaly detection in IoT systems through sensor association". In: ***ICCAD***. 2020, pp. 1–9.

[123] Harsha Kumara Kalutarage, M. Omar Al-Kadri, Madeline Cheah, and Garikayi Madzudzo. "Context-aware Anomaly Detector for Monitoring Cyber Attacks on Automotive CAN Bus". In: ***ACM Computer Science in Cars Symposium***. 2019, pp. 1–8. DOI: 10.1145/3359999.3360496.

[124] Mengjie Zhao and Olga Fink. "DyEdgeGAT: Dynamic Edge via Graph Attention for Early Fault Detection in IIoT Systems". In: ***arXiv preprint arXiv:2307.03761*** (2023).

[125] Mohammad Saidur Rahman, MAP Chamikara, Ibrahim Khalil, and Abdelaziz Bouras. "Blockchain-of-blockchains: An interoperable blockchain platform for ensuring IoT data integrity in smart city". In: ***Journal of Industrial Information Integration*** 30 (2022), p. 100408.

[126] Shanshan Zhao, Shancang Li, and Yufeng Yao. "Blockchain enabled industrial Internet of Things technology". In: ***IEEE Transactions on Computational Social Systems*** 6.6 (2019), pp. 1442–1453.

[127] Arvind Narayanan. ***Bitcoin and cryptocurrency technologies: a comprehensive introduction***. Princeton University Press, 2016.

[128] Sabah Suhail, Rasheed Hussain, Abid Khan, and Choong Seon Hong. "On the role of hash-based signatures in quantum-safe internet of things: Current solutions and future directions". In: ***IEEE Internet of Things Journal*** 8.1 (2020), pp. 1–17.

[129] Du Mingxiao, Ma Xiaofeng, Zhang Zhe, Wang Xiangwei, and Chen Qijun. "A review on consensus algorithm of blockchain". In: ***2017 IEEE international conference on systems, man, and cybernetics (SMC)***. IEEE. 2017, pp. 2567–2572.

[130] Julien Polge, Jérémy Robert, and Yves Le Traon. "Permissioned blockchain frameworks in the industry: A comparison". In: ***Ict Express*** 7.2 (2021), pp. 229–233.

[131] Wenyu Li, Chenglin Feng, Lei Zhang, Hao Xu, Bin Cao, and Muhammad Ali Imran. "A scalable multi-layer PBFT consensus for blockchain". In: ***IEEE Transactions on Parallel and Distributed Systems*** 32.5 (2020), pp. 1146–1160.

[132] Diego Ongaro and John Ousterhout. "The raft consensus algorithm". In: ***Lecture Notes CS*** 190 (2015), p. 2022.

[133] Ali Vatankhah Barenji, Zhi Li, and Wai Ming Wang. "Blockchain cloud manufacturing: Shop floor and machine level". In: ***Smart SysTech 2018; European conference on smart objects, systems and technologies***. VDE. 2018, pp. 1–6.

[134] Michael Nofer, Peter Gomber, Oliver Hinz, and Dirk Schiereck. "Blockchain". In: ***Business & Information Systems Engineering*** 59 (2017), pp. 183–187.

[135] Eugenia Politou, Fran Casino, Efthimios Alepis, and Constantinos Patsakis. "Blockchain mutability: Challenges and proposed solutions". In: ***IEEE Transactions on Emerging Topics in Computing*** 9.4 (2019), pp. 1972–1986.

[136] Ray Y Zhong, Xun Xu, Eberhard Klotz, and Stephen T Newman. "Intelligent manufacturing in the context of industry 4.0: a review". In: ***Engineering*** 3.5 (2017), pp. 616–630.

[137] Zhi Li, Layne Liu, Ali Vatankhah Barenji, and Waiming Wang. "Cloud-based manufacturing blockchain: Secure knowledge sharing for injection mould redesign". In: ***Procedia Cirp*** 72 (2018), pp. 961–966.

[138] Jiachi Chen, Xin Xia, David Lo, John Grundy, Xiapu Luo, and Ting Chen. "Defining smart contract defects on ethereum". In: ***IEEE Transactions on Software Engineering*** 48.1 (2020), pp. 327–345.

[139] Shimin Liu, Yuqian Lu, Jie Li, Xingwang Shen, Xuemin Sun, and Jinsong Bao. "A blockchain-based interactive approach between digital twin-based manufacturing systems". In: ***Computers & Industrial Engineering*** 175 (2023), p. 108827.

[140] Thomas Kobzan, Alexander Biendarra, Sebastian Schriegel, Thomas Herbst, Thomas Müller, and Jürgen Jasperneite. "Utilizing blockchain technology in industrial manufacturing with the help of network simulation". In: ***2018 IEEE 16th International Conference on Industrial Informatics (INDIN)***. IEEE. 2018, pp. 152–159.

[141] Philipp Schmid, Alisa Schaffhäuser, and Rasha Kashef. "IoTBChain: Adopting Blockchain Technology to Increase PLC Resilience in an IoT Environment". In: ***Information*** 14.8 (2023), p. 437.

[142] Abdul Jabbar and Samir Dani. "Investigating the link between transaction and computational costs in a blockchain environment". In: ***International Journal of Production Research*** 58.11 (2020), pp. 3423–3436.

[143] Denis Stefanescu, Leticia Montalvillo, Patxi Galán-García, Juanjo Unzilla, and Aitor Urbieta. "A Systematic Literature Review of Lightweight Blockchain for IoT". In: ***IEEE Access*** (2022).

[144] Salaheddine Kably, Mounir Arioua, and Nabih Alaoui. "Lightweight blockchain network architecture for IoT devices". In: ***2020 International Symposium on Advanced Electrical and Communication Technologies (ISAECT)***. IEEE. 2020, pp. 1–6.

[145] Omeshika AS Ekanayake and Malka N Halgamuge. "Lightweight blockchain framework using enhanced master-slave blockchain paradigm: Fair rewarding mechanism using reward accuracy model". In: ***Information Processing & Management*** 58.3 (2021), p. 102523.

[146] Yinqiu Liu, Kun Wang, Yun Lin, and Wenyao Xu. "LightChain: a lightweight blockchain system for industrial internet of things". In: ***IEEE Transactions on Industrial Informatics*** 15.6 (2019), pp. 3571–3581.

[147] Safiullah Khan, Wai-Kong Lee, and Seong Oun Hwang. "AEchain: A lightweight blockchain for IoT applications". In: ***IEEE Consumer Electronics Magazine*** 11.2 (2021), pp. 64–76.

[148] Eranga Bandara, Deepak Tosh, Peter Foytik, Sachin Shetty, Nalin Ranasinghe, and Kasun De Zoysa. "Tikiri—Towards a lightweight blockchain for IoT". In: ***Future Generation Computer Systems*** 119 (2021), pp. 154–165.

[149] Samia Yasmin, Md Faruk Abdullah Al Sohan, Md Navid Bin Anwar, Mehedi Hasan, and GM Farhad Hossain. "SFC: a lightweight blockchain model for smart food industry". In: ***2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)***. IEEE. 2021, pp. 699–703.

[150] Ali Haleem Alkhazaali and ATA Oğuz. "Lightweight fog based solution for privacy-preserving in IoT using blockchain". In: ***2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)***. IEEE. 2020, pp. 1–10.

[151] Fengqun Wang, Jie Cui, Qingyang Zhang, Debiao He, Chengjie Gu, and Hong Zhong. "Lightweight and Secure Data Sharing Based On Proxy Re-Encryption for Blockchain-Enabled Industrial Internet of Things". In: ***IEEE Internet of Things Journal*** (2023).

[152] Dongjun Na and Sejin Park. "IoT-chain and monitoring-chain using multi-level blockchain for IoT security". In: ***Sensors*** 22.21 (2022), p. 8271.

[153] Xiaohai Dai, Zhaonan Zhang, Jiang Xiao, Jingtao Yue, Xia Xie, and Hai Jin. "Gradeddag: An asynchronous dag-based bft consensus with lower latency". In: ***2023 42nd International Symposium on Reliable Distributed Systems (SRDS)***. IEEE. 2023, pp. 107–117.

[154] Xiaohai Dai, Guanxiong Wang, Jiang Xiao, Zhengxuan Guo, Rui Hao, Xia Xie, and Hai Jin. "Lightdag: A low-latency dag-based bft consensus through lightweight broadcast". In: ***Cryptology ePrint Archive*** (2024).

[155] Balaji Arun, Zekun Li, Florian Suri-Payer, Sourav Das, and Alexander Spiegelman. "Shoal++: High Throughput DAG BFT Can Be Fast!" In: ***arXiv preprint arXiv:2405.20488*** (2024).

[156] Alexander Spiegelman, Neil Giridharan, Alberto Sonnino, and Lefteris Kokoris-Kogias. "Bullshark: Dag bft protocols made practical". In: ***Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security***. 2022, pp. 2705–2718.

[157] Dominik Kempa and Tomasz Kociumaka. "Collapsing the hierarchy of compressed data structures: Suffix arrays in optimal compressed space". In: ***2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)***. IEEE. 2023, pp. 1877–1886.

[158] Natalia Chaudhry and Muhammad Murtaza Yousaf. "Consensus algorithms in blockchain: Comparative analysis, challenges and opportunities". In: ***2018 12th international conference on open source systems and technologies (ICOSST)***. IEEE. 2018, pp. 54–63.

[159] Maike Jansen, Tobias Meisen, Christiane Plociennik, Holger Berg, André Pomp, and Waldemar Windholz. ***Stop Guessing in the Dark: Identified Requirements for Digital Product Passport Systems in: Systems 11, 123***. 2023.

[160] Lukas Stratmann, Gerrit Hoeborn, Christoph Pahl, and Günther Schuh. "Classification of product data for a Digital Product Passport in the manufacturing industry". In: (2023).

[161] Thomas Adisorn, Lena Tholen, and Thomas Götz. "Towards a digital product passport fit for contributing to a circular economy". In: ***Energies*** 14.8 (2021), p. 2289.

[162] Joerg Walden, Angelika Steinbrecher, and Maroye Marinkovic. "Digital product passports as enabler of the circular economy". In: ***Chemie Ingenieur Technik*** 93.11 (2021), pp. 1717–1727.

[163] Julia V Donetskaya and Yuriy A Gatchin. "Development of requirements for the content of a digital passport and design solutions". In: ***Journal of Physics: Conference Series***. Vol. 1828. 1. IOP Publishing. 2021, p. 012102.

[164] Szymon Nowacki, Gokay Meric Sisik, and Constantinos Marios Angelopoulos. "Digital Product Passports: Use Cases Framework and Technical Architecture Using DLT and Smart Contracts". In: ***2023 19th International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT)***. IEEE. 2023, pp. 373–380.

[165] Arafat Saleheen and Shafin Afrid. ***Potential of decentralised blockchains for the digital product passport: Need for traceability and transparency in textile industries***. 2023.

[166] Fabian Falco. "Distributed ledger technology in the circular economy: Enable traceability and transparency of recyclable products with a digital product passport platform". PhD thesis. FH Vorarlberg (Fachhochschule Vorarlberg).

[167] Konstantinos Voulgaridis, Thomas Lagkas, Constantinos Marios Angelopoulos, Alexandros Apostolos A Boulogeorgos, Vasileios Argyriou, and Panagiotis Sarigiannidis. "Digital product passports as enablers of digital circular economy: a framework based on technological perspective". In: ***Authorea Preprints*** (2023), pp. 1–17.

[168] Louise Axon, Katherine Fletcher, Arianna Schuler Scott, Marcel Stolz, Robert Hannigan, Ali El Kaafarani, Michael Goldsmith, and Sadie Creese. "Emerging cybersecurity capability gaps in the industrial internet of things: Overview and research agenda". In: ***Digital Threats: Research and Practice*** 3.4 (2022), pp. 1–27.

[169] Hakan Kayan, Matthew Nunes, Omer Rana, Pete Burnap, and Charith Perera. "Cybersecurity of industrial cyber-physical systems: A review". In: ***ACM Computing Surveys (CSUR)*** 54.11s (2022), pp. 1–35.

[170] Adel Alqudhaibi, Majed Albarrak, Sandeep Jagtap, Nikki Williams, and Konstantinos Salonitis. "Securing industry 4.0: Assessing cybersecurity challenges and proposing strategies for manufacturing management". In: ***Cyber Security and Applications*** 3 (2025), p. 100067.

[171] Nteziriza Nkerabahizi Josbert, Min Wei, Wang Ping, and Ahsan Rafiq. "A look into smart factory for Industrial IoT driven by SDN technology: A comprehensive survey of taxonomy, architectures, issues and future research orientations". In: ***Journal of King Saud University-Computer and Information Sciences*** (2024), p. 102069.

[172] Linah Khamis Aljaryan, Wasan Hussein Alfalahi, and Thamer Saleh Al Khamis. "Cyberattacks and Solutions for Future Factories". In: ***2022 14th International Conference on Computational Intelligence and Communication Networks (CICN)***. IEEE. 2022, pp. 1–7.

[173] Hamed Sarjan, Amir Ameli, and Mohsen Ghafouri. "Cyber-security of industrial internet of things in electric power systems". In: ***IEEE Access*** 10 (2022), pp. 92390–92409.

[174] Rafiullah Khan, Kieran McLaughlin, David Laverty, and Sakir Sezer. "STRIDE-based threat modeling for cyber-physical systems". In: ***2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)***. IEEE. 2017, pp. 1–6.

[175] OWASP Foundation. ***OWASP Threat Dragon***. https://owasp.org/www-project-threat-dragon/. Accessed: 2025-02-13.

[176] Cristina Regueiro, Iván Gutierrez-Agüero, Sergio Anguita, Santiago de Diego, and Oscar Lage. "Protocol for identity management in industrial IoT based on hyperledger Indy". In: ***Int. J. Com. Dig. Sys*** 12.1 (2022).

[177] Akanksha Dixit, Max Smith-Creasey, and Muttukrishnan Rajarajan. "A decentralized IIoT identity framework based on self-sovereign identity using blockchain". In: ***2022 IEEE 47th Conference on Local Computer Networks (LCN)***. IEEE. 2022, pp. 335–338.

[178] Syrine Sahmim, Hamza Gharsellaoui, and Sadok Bouamama. "Edge computing: smart identity wallet based architecture and user centric". In: ***Procedia Computer Science*** 159 (2019), pp. 1246–1257.

[179] Marius Popa, Sebastian Michael Stoklossa, and Somnath Mazumdar. "Chaindiscipline-towards a blockchain-iot-based self-sovereign identity management framework". In: ***IEEE Transactions on Services Computing*** 16.5 (2023), pp. 3238–3251.

[180] Parikshit N Mahalle and Poonam N Railkar. ***Identity management for internet of things***. CRC Press, 2022.

[181] Sebastian Gajek, Hans Löhr, Ahmad-Reza Sadeghi, and Marcel Winandy. "TruWallet: trustworthy and migratable wallet-based web authentication". In: ***Proceedings of the 2009 ACM workshop on Scalable trusted computing***. 2009, pp. 19–28.

[182] Mirko Forti. "A Legal Identity for All Through Artificial Intelligence: Benefits and Drawbacks in Using AI Algorithms to Accomplish SDG 16.9". In: ***The Ethics of Artificial Intelligence for the Sustainable Development Goals***. Springer, 2023, pp. 253–267.

[183] Konstantinos Lampropoulos, Nikos Kyriakoulis, and Spyros Denazis. "Identity Management through a global Discovery System based on Decentralized Identities". In: ***arXiv preprint arXiv:2212.02185*** (2022).

[184] R Vijaya Manikandan, K Gurunathan, D Ravindran, M Sanjai, and VP Pranav Raja. "An Novel Algorithm for Cloud Secure Storage Using Cloud Dispersion and Block Chain System". In: ***2023 4th International Conference on Signal Processing and Communication (ICSPC)***. IEEE. 2023, pp. 372–376.

[185] Syreen Banabilah, Moayad Aloqaily, Eitaa Alsayed, Nida Malik, and Yaser Jararweh. "Federated learning review: Fundamentals, enabling technologies, and future applications". In: ***Information processing & management*** 59.6 (2022), p. 103061.

[186] Boyuan Gao, Hairong Yan, and Rui Tian. "A privacy-aware cross-domain device authentication scheme for IIoT based on blockchain". In: ***2021 IEEE 23rd Int Conf on High Performance Computing & Communications; 7th Int Conf on Data Science & Systems; 19th Int Conf on Smart City; 7th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys)***. IEEE. 2021, pp. 561–570.

[187] Jie Cui, Fengqun Wang, Qingyang Zhang, Chengjie Gu, and Hong Zhong. "Efficient batch authentication scheme based on edge computing in iiot". In: ***IEEE Transactions on Network and Service Management*** 20.1 (2022), pp. 357–368.

[188] Fengqun Wang, Jie Cui, Qingyang Zhang, Debiao He, Chengjie Gu, and Hong Zhong. "Blockchain-based lightweight message authentication for edge-assisted cross-domain industrial internet of things". In: ***IEEE transactions on dependable and secure computing*** (2023).

[189] Khalid Mahmood, Salman Shamshad, Muhammad Asad Saleem, Rupak Kharel, Ashok Kumar Das, Sachin Shetty, and Joel JPC Rodrigues. "Blockchain and PUF-based secure key establishment protocol for cross-domain digital twins in industrial Internet of Things architecture". In: ***Journal of Advanced Research*** 62 (2024), pp. 155–163.

[190] Jayasree Sengupta, Sushmita Ruj, and Sipra Das Bit. "A comprehensive survey on attacks, security issues and blockchain solutions for IoT and IIoT". In: ***Journal of Network and Computer Applications*** 149 (2020), p. 102481.

[191] Chi Hieu Le, Dang Thang Le, Daniel Arey, Popan Gheorghe, Anh My Chu, Xuan Bien Duong, Trung Thanh Nguyen, Trong Toai Truong, Chander Prakash, Shi-Tian Zhao, et al. "Challenges and conceptual framework to develop heavy-load manipulators for smart factories". In: ***International Journal of Mechatronics and Applied Mechanics*** 8.2 (2020), pp. 209–216.

[192] Fatemeh Ghovanlooy Ghajar, Axel Sikora, and Dominik Welte. "Schloss: Blockchain-based system architecture for secure industrial iot". In: ***Electronics*** 11.10 (2022), p. 1629.

[193] Salabat Khan, Fei Luo, Zijian Zhang, Farhan Ullah, Farhan Amin, Syed Furqan Qadri, Md Belal Bin Heyat, Rukhsana Ruby, Lu Wang, Shamsher Ullah, et al. "A survey on X. 509 public-key infrastructure, certificate revocation, and their modern implementation on blockchain and ledger technologies". In: ***IEEE Communications Surveys & Tutorials*** 25.4 (2023), pp. 2529–2568.

[194] Rolf Oppliger. ***SSL and TLS: Theory and Practice***. Artech House, 2023.

[195] Alireza Esfahani, Georgios Mantas, Rainer Matischek, Firooz B Saghezchi, Jonathan Rodriguez, Ani Bicaku, Silia Maksuti, Markus G Tauber, Christoph Schmittner, and Joaquim Bastos. "A lightweight authentication mechanism for M2M communications in industrial IoT environment". In: ***IEEE Internet of Things Journal*** 6.1 (2017), pp. 288–296.

[196] Nir Kshetri. "Blockchain's roles in strengthening cybersecurity and protecting privacy". In: ***Telecommunications policy*** 41.10 (2017), pp. 1027–1038.

[197] Emiliano Sisinni, Abusayeed Saifullah, Song Han, Ulf Jennehag, and Mikael Gidlund. "Industrial internet of things: Challenges, opportunities, and directions". In: ***IEEE transactions on industrial informatics*** 14.11 (2018), pp. 4724–4734.

[198] Abdulmalik Humayed, Jingqiang Lin, Fengjun Li, and Bo Luo. "Cyber-physical systems security—A survey". In: ***IEEE Internet of Things Journal*** 4.6 (2017), pp. 1802–1831.

[199] Mohiuddin Ahmed, Abdun Naser Mahmood, and Jiankun Hu. "A survey of network anomaly detection techniques". In: ***Journal of Network and Computer Applications*** 60 (2016), pp. 19–31.

[200] Raghavendra Chalapathy and Sanjay Chawla. "Deep learning for anomaly detection: A survey". In: ***arXiv preprint arXiv:1901.03407*** (2019).

[201] Guansong Pang, Longbing Cao, and Charu Aggarwal. "Deep learning for anomaly detection: Challenges, methods, and opportunities". In: ***Proceedings of the 14th ACM international conference on web search and data mining***. 2021, pp. 1127–1130.

[202] Yonatan Amaru, Prasanna Wudali, Yuval Elovici, and Asaf Shabtai. "RAPID: Robust APT detection and investigation using context-aware deep learning". In: ***arXiv preprint arXiv:2406.05362*** (2024).

[203] Sri Harsha Mekala, Zubair Baig, Adnan Anwar, and Sherali Zeadally. "Cybersecurity for Industrial IoT (IIoT): Threats, countermeasures, challenges and future directions". In: ***Computer Communications*** 208 (2023), pp. 294–320. ISSN: 0140-3664. DOI: 10.1016/j.comcom.2023.06.020.

[204] Scott Lundberg. "A unified approach to interpreting model predictions". In: ***arXiv preprint arXiv:1705.07874*** (2017).

[205] Leo Breiman. "Random forests". In: ***Machine learning*** 45 (2001), pp. 5–32.

[206] Mariana Belgiu and Lucian Drăguţ. "Random forest in remote sensing: A review of applications and future directions". In: ***ISPRS journal of photogrammetry and remote sensing*** 114 (2016), pp. 24–31.

[207] Christoph Molnar. ***Interpretable machine learning***. Lulu. com, 2020.

[208] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. "Optuna: A next-generation hyperparameter optimization framework". In: ***Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining***. 2019, pp. 2623–2631.

[209] Vinod Nair and Geoffrey E Hinton. "Rectified linear units improve restricted boltzmann machines". In: ***Proceedings of the 27th international conference on machine learning (ICML-10)***. 2010, pp. 807–814.

[210] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. ***The elements of statistical learning: data mining, inference, and prediction***. Vol. 2. Springer, 2009.

[211] Alaa Tharwat, Tarek Gaber, Abdelhameed Ibrahim, and Aboul Ella Hassanien. "Linear discriminant analysis: A detailed tutorial". In: ***AI communications*** 30.2 (2017), pp. 169–190.

[212] Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim. "Do we need hundreds of classifiers to solve real world classification problems?" In: ***The journal of machine learning research*** 15.1 (2014), pp. 3133–3181.

[213] ***Intrusion Detection Evaluation Dataset (CIC-IDS2017)***. https://www.unb.ca/cic/datasets/ids-2017.html.

[214] Jesse Davis and Mark Goadrich. "The relationship between Precision-Recall and ROC curves". In: ***Proceedings of the 23rd international conference on Machine learning***. 2006, pp. 233–240.

[215] F. Stodt. ***AE-LDA***. GitHub repository. [Online; accessed 2-April-2024]. 2023. URL: https://github.com/f11691/Behaviour_Anomaly_Detector.

[216] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V Chawla. "Heterogeneous graph neural network". In: ***Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining***. 2019, pp. 793–803.

[217] Jörg Reichardt and Stefan Bornholdt. "Statistical mechanics of community detection". In: ***Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*** 74.1 (2006), p. 016110.

[218] Pascal Pons and Matthieu Latapy. "Computing communities in large networks using random walks". In: ***Computer and Information Sciences-ISCIS 2005: 20th International Symposium, Istanbul, Turkey, October 26-28, 2005. Proceedings 20***. Springer. 2005, pp. 284–293.

[219] Sayan Ghosh, Mahantesh Halappanavar, Antonino Tumeo, Ananth Kalyanaraman, Hao Lu, Daniel Chavarria-Miranda, Arif Khan, and Assefaw Gebremedhin. "Distributed louvain algorithm for graph community detection". In: ***IPDPS***. IEEE. 2018, pp. 885–895.

[220] Martin Rosvall, Daniel Axelsson, and Carl T Bergstrom. "The map equation". In: ***The European Physical Journal Special Topics*** 178.1 (2009), pp. 13–23.

[221] Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. "Near linear time algorithm to detect community structures in large-scale networks". In: ***Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*** 76.3 (2007), p. 036106.

[222] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. "Heterogeneous graph attention network". In: ***The world wide web conference***. 2019, pp. 2022–2032.

[223] Thomas N Kipf and Max Welling. "Semi-supervised classification with graph convolutional networks". In: ***arXiv preprint arXiv:1609.02907*** (2016).

[224] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. "Focal loss for dense object detection". In: ***Proceedings of the IEEE international conference on computer vision***. 2017, pp. 2980–2988.

[225] Mohanad Sarhan, Siamak Layeghy, and Marius Portmann. "Evaluating standard feature sets towards increased generalisability and explainability of ML-based network intrusion detection". In: ***Big Data Research*** 30 (2022), p. 100359.

[226] Iman Sharafaldin, Arash Habibi Lashkari, Ali A Ghorbani, et al. "Toward generating a new intrusion detection dataset and intrusion traffic characterization." In: ***ICISSp*** 1 (2018), pp. 108–116.

[227] Xinlei Wang, Xiaojuan Wang, Mingshu He, Min Zhang, and Zikui Lu. "Spatial-temporal graph model based on attention mechanism for anomalous IoT intrusion detection". In: ***IEEE Transactions on Industrial Informatics*** (2023).

[228] Guanghan Duan, Hongwu Lv, Huiqiang Wang, and Guangsheng Feng. "Application of a dynamic line graph neural network for intrusion detection with semisupervised learning". In: ***IEEE Transactions on Information Forensics and Security*** 18 (2022), pp. 699–714.

[229] Dinh-Hau Tran and Minho Park. "FN-GNN: A novel graph embedding approach for enhancing graph neural networks in network intrusion detection systems". In: ***Applied Sciences*** 14.16 (2024), p. 6932.

[230] Qingfeng Ding and Jinguo Li. "AnoGLA: An efficient scheme to improve network anomaly detection". In: ***Journal of Information Security and Applications*** 66 (2022), p. 103149.

[231] Cristina Alcaraz. "Secure interconnection of IT-OT networks in industry 4.0". In: ***Critical Infrastructure Security and Resilience: Theories, Methods, Tools and Technologies*** (2019), pp. 201–217.

[232] Tiago M Fernandez-Carames and Paula Fraga-Lamas. "Towards post-quantum blockchain: A review on blockchain cryptography resistant to quantum computing attacks". In: ***IEEE access*** 8 (2020), pp. 21091–21116.

[233] Mohammed BM Kamel, Yuping Yan, Peter Ligeti, and Christoph Reich. "Attribute Verifier for Internet of Things". In: ***2022 32nd International Telecommunication Networks and Applications Conference (ITNAC)***. IEEE. 2022, pp. 1–3.

[234] Mohammed BM Kamel, Peter Ligeti, and Christoph Reich. "D3vn: Decentralized abe-based distributed data validation network". In: ***Proceedings of Seventh International Congress on Information and Communication Technology: ICICT 2022, London, Volume 4***. Springer. 2022, pp. 653–661.

[235] Mohammed BM Kamel, Wisam Dawood Abdullah, Alaa Khalaf Hamoud, Dalton CG Valadares, Ammar Shareiyat, and Peter Ligeti. "3L-AODV: Three Layer Security Protocol for Grayhole Attack Mitigation in MANET". In: ***International Congress on Information and Communication Technology***. Springer. 2023, pp. 813–823.

[236] Joseph A Akinyele, Christina Garman, Ian Miers, Matthew W Pagano, Michael Rushanan, Matthew Green, and Aviel D Rubin. "Charm: a framework for rapidly prototyping cryptosystems". In: ***Journal of Cryptographic Engineering*** 3 (2013), pp. 111–128.

[237] Jusik Yun, Yunyeong Goh, and Jong-Moon Chung. "Analysis of mining performance based on mathmatical approach of PoW". In: ***2019 International conference on electronics, information, and communication (ICEIC)***. IEEE. 2019, pp. 1–2.

[238] Bin Cao, Zhenghui Zhang, Daquan Feng, Shengli Zhang, Lei Zhang, Mugen Peng, and Yun Li. "Performance analysis and comparison of PoW, PoS and DAG based blockchains". In: ***Digital Communications and Networks*** 6.4 (2020), pp. 480–485.

[239] Harish Sukhwani, José M Martínez, Xiaolin Chang, Kishor S Trivedi, and Andy Rindos. "Performance modeling of PBFT consensus process for permissioned blockchain network (hyperledger fabric)". In: ***2017 IEEE 36th symposium on reliable distributed systems (SRDS)***. IEEE. 2017, pp. 253–255.

[240] Char Sample, Cragin Shelton, Sin Ming Loo, Connie Justice, Lynette Hornung, and Ian Poynter. "ZTA: Never Trust, Always Verify". In: ***ECCWS 2022 21st European Conference on Cyber Warfare and Security***. Academic Conferences and publishing limited. 2022.

[241] Fatemeh Stodt, Christoph Reich, and Fabrice Theoleyre. "Context-Aware Anomaly Detection by Community Detection in the Internet of Things". Manuscript submitted for publication. 2024.

[242] Blake E. Strom, Andy Applebaum, Doug P. Miller, Kathryn C. Nickels, Adam G. Pennington, and Cody B. Thomas. ***MITRE ATT&CK®: Design and Philosophy***. Tech. rep. MP180360R1. MITRE, 2020.

[243] F. Stodt. ***ZTA***. GitHub repository. [Online; accessed 19-November-2024]. 2024. URL: https://github.com/f11691/ZTA2.

[244] Binanda Sengupta and Anantharaman Lakshminarayanan. "Distritrust: Distributed and low-latency access validation in zero-trust architecture". In: ***Journal of Information Security and Applications*** 63 (2021), p. 103023.

[245] Haoran Xie, Yujue Wang, Yong Ding, Changsong Yang, Hai Liang, and Bo Qin. "Industrial Wireless Internet Zero Trust Model: Zero Trust Meets Dynamic Federated Learning with Blockchain". In: ***IEEE Wireless Communications*** 31.2 (2024), pp. 22–29.

[246] Matteo Podrecca, Giovanna Culot, Guido Nassimbeni, and Marco Sartor. "Information security and value creation: The performance implications of ISO/IEC 27001". In: ***Computers in Industry*** 142 (2022), p. 103744.

[247] Lucas Franke, Huayu Liang, Sahar Farzanehpour, Aaron Brantly, James C Davis, and Chris Brown. "An exploratory mixed-methods study on general data protection regulation (gdpr) compliance in open-source software". In: ***Proceedings of the 18th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement***. 2024, pp. 325–336.

[248] Rolf Moulton and Robert S Coles. "Applying information security governance". In: ***Computers & Security*** 22.7 (2003), pp. 580–584.

[249] Adil Khan, Ar Junejo, M Naeem, M Sattar, and AH Malik. "Inter-organizational cloud computing and robust scalability in current scenario and beyond". In: ***Automatic Control and Computer Sciences*** 56.1 (2022), pp. 26–37.

[250] Vivek Kumar Prasad, Debabrata Dansana, Madhuri D Bhavsar, Biswaranjan Acharya, Vassilis C Gerogiannis, and Andreas Kanavos. "Efficient resource utilization in IoT and cloud computing". In: ***Information*** 14.11 (2023), p. 619.