A Critical Overview of Industrial Internet of Things Security and Privacy Issues Using a Layer-Based Hacking Scenario

Anselme Herman Eyeleko and Tao Feng

***Abstract*—We have witnessed significant technological advancement over the past few years, including the Internet of Things (IoT). The IoT's ability to connect consumer appliances to the Internet has changed the way we live. As a result of the significant benefits that IoT has brought to household usage, it has become a topic of discussion for research departments, leading to its expansion into industrial sectors, commonly known as the Industrial IoT (IIoT). IIoT enables automation and the use of intelligent machines to improve product manufacturing processes and enhance our lives as customers. However, as IIoT-enabling technology and applications continue to grow, security issues and privacy protection challenges become harder to manage, which frequently results in data breaches and sensitive information disclosures. This article first explains what the reader needs to know about the IIoT system architecture in Industry 4.0 to make it easier for them to understand. Second, a hacking scenario is utilized as a methodology to conduct an in-depth analysis of various security issues, as well as their impacts and countermeasures, for each level of the IIoT architecture. Additionally, our hacking scenarios present a variety of targets from which malicious actors can launch their assaults. Third, we provide a thorough review of the various blockchain solutions currently being employed to protect IIoT systems. Finally, this article draws to a close by outlining certain gaps and potential solutions that could be investigated in subsequent studies to strengthen security and enhance privacy for IIoT systems.**

***Index Terms*—Blockchain, cybersecurity, hacking scenarios, Industrial Internet of Things (IIoT), security and privacy.**

I. Introduction

THE INDUSTRIAL Internet of Things (IIoT) has become more prevalent in industrial sectors as a result of the rapid development of cloud computing, artificial intelligence (AI), big data analytics, the Internet of Things (IoT), and other emerging technologies, along with digital transformation mechanisms, have fundamentally altered how the industry responds to societal needs [1], [2], [3].

The IIoT [4] starts with networked instruments, sensors, and several other devices spread across the factory's production unit [5], [6], [7]. They interact with computer-driven industrial applications and are networked together. IIoT systems can assist in the real-time monitoring of conveyor belt production flow [8]. All of the sensor-collected data is transmitted to the cloud for analysis and the development of predictive models that assist engineers in implementing condition-based maintenance alerts to reduce machine downtime and boost production by anticipating failure. Additionally, it can remotely control equipment and modify other settings to save energy and reduce costs. The research and development (R&D) divisions of businesses can get insights from the equipment failures and usage patterns of customers, allowing businesses to re-engineer goods to improve quality. Employees within the company can use IIoT dashboards to access data and uncover valuable intelligence. Smart manufacturing is made possible by the IIoT, which combines sensors with data analytics, cloud-based infrastructure, and visualization [9]. Also referred to as IT/OT convergence, IIoT is the fusion of informational and operational technology. Fig. 1 illustrates how IIoT enables smart manufacturers to enhance efficiency, boost productivity, and reduce costs throughout the supply chain.

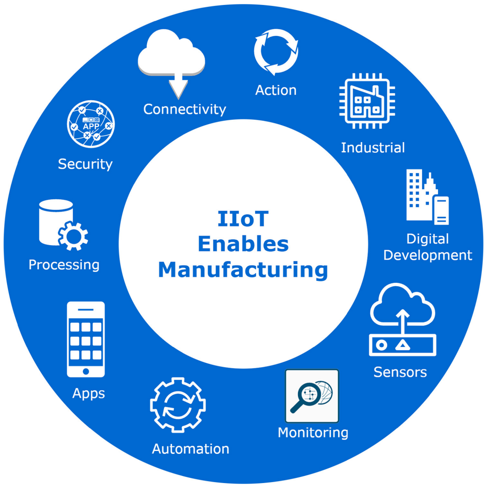


Fig. 1. IIoT in smart manufacturing.

In the traditional cyber-ecosystem, businesses rely on hardware, software, and systems to access information. When IT interacts with OT environments, forming a "cyber-physical system" (CPS) in sectors like manufacturing, it exposes IIoT architecture to cyber-physical threats at every layer [10], [11]. With the increasing adoption of IIoT technologies, security

Manuscript received 30 October 2022; revised 25 July 2023; accepted 20 August 2023. Date of publication 24 August 2023; date of current version 7 December 2023. This work was supported by the National Natural Science Foundation of China under Grant 62162039 and Grant 61762060. *(Corresponding author: Tao Feng.)*

The authors are with the School of Computer Communication, Lanzhou University of Technology, Lanzhou 730050, China (e-mail: fengt@lut.edu.cn).

Digital Object Identifier 10.1109/JIOT.2023.3308195

and privacy concerns escalate, leading to information leakage, sensitive data disclosure, and data theft [12], [13].

In industrial engineering projects, budgets are crucial, leading businesses to avoid investing in costly machinery like industrial machines, human–machine interfaces (HMIs), or programmable logic controllers (PLCs) to save money. However, relying on outdated central processing units (CPUs) and inadequate technology may compromise security against cyber threats like ransomware, malware, data breaches, or server takeovers due to insufficient processing power [14].

The fact that industrial edge devices might be purchased from various manufacturers, thus creating a mixed vendor environment, is another problem. Because different devices may have varying levels of protection, this complicates efforts to enforce security standards. Lack of patch management is one of the main causes of inherent IIoT security vulnerabilities [15]. Even though practically all devices receive security updates regularly, some users and companies might postpone installing patches. This delay increases the risk of security threats and data breaches, potentially exposing sensitive personal information. This was perfectly illustrated by the WannaCry ransomware attack [16], where despite the fact that Microsoft published a patch to fix the vulnerability, several companies in the machine and industrial networking industries did not successfully install it. Over 200 000 computing devices have been affected, with billions of dollars in collateral damage.

Industrial commercials frequently encounter the issue of insecure network services due to the use of less secure network protocols. Neglecting to change default user IDs and passwords or enabling full-scale encryption (HTTPS/SSL) on devices can lead to cyber threats, allowing hackers to decrypt traffic and gain access to authentication information.

Another concern is the lack of privacy protection in industrial companies. They collect vast amounts of sensitive data from devices, sensors, and machinery, sending it to third-party cloud services for analysis. This data can cause serious harm if released. Therefore, ensuring the protection of consumers' privacy when publishing or sharing their information has become a top priority for many IIoT companies to maintain the smooth and efficient operation of their businesses.

The issues raised above clearly demonstrate the need for proactive defensive mechanisms and policies to be established to increase the security of the IIoT system while assuring the protection of workers' and consumers' privacy. Several research projects and solutions have already been put forth to strengthen the security and privacy protection of IIoT systems at the edge of devices, networks, applications, and the cloud. However, most of these investigations do not enfold all feasible security and privacy vulnerabilities, threats, and countermeasures at each layer of the IIoT architecture, which is the main reason that motivates us to write this review. To properly comprehend why security and privacy concerns continue to be a substantial barrier to IIoT adoption, this article provides a comprehensive overview of the literature on security and privacy issues in the IIoT. The main contributions of this work are listed below.

1) We first provide the reader with the essential background he needs to understand IIoT in Industry 4.0, as well as some basic definitions, before developing a taxonomy to organize the entire layered modular architecture of IIoT, including the perception layer, network layer, processing layer, and application layer.
2) Second, we carry out an extensive analysis of potential privacy and security threats for each level of the IIoT framework. A hacking scenario is used as a methodology to fully describe and illustrate the entire process of how an adversary gathers knowledge about system flaws and launches attack vectors. The targets from which the hacker can launch his attacks are also provided in our hacking scenarios. The hypothetical hacking scenario considers all potential outcomes, impacts, and defenses against cyberattacks.
3) Third, we offer a thorough analysis of current blockchain-based IIoT privacy solutions. The study comes to a close by discussing gaps and alternative solutions that future research should consider in order to improve the security and privacy of IIoT systems.

The remainder of this article is organized as follows. Section II describes the principal concepts of IIoT. Section III includes a layer-based hacking scenario for investigating IIoT vulnerabilities, security risks, and defenses. Section IV discusses various blockchain-based privacy protection strategies for IIoT systems. A discussion and an analogy with recent surveys are presented in Section V. Then, in Section VI, we outline open issues and suggest innovative strategies and approaches for research direction. Finally, Section VII concludes this article.

TABLE I
LIST OF ACRONYMS AND ABBREVIATIONS

Acronym	*Description*
CPS	Cyber-Physical System
HMI	Human-Machine Interface
PLC	Programmable Logic Controller
ERP	Enterprise Resource Planning
CRM	Customer Relationship Management
SCADA	Supervisory Control And Data Acquisition
MES	Manufacturing Execution System
WMS	Warehouse Management System
ICS	Industrial Control System
MSC	Manufacturing Service Collaboration
DoS	Denial-of-Service
DDoS	Dstributed Denial-of-Service
MITM	Man-In-The-Middle
IDS	Intrusion Detection Systems
IPS	Intrusion prevention systems
MFA	Multi-Factor Authentication
CTH	Cyber Threat Hunting
SQL	Structured Query Language
LDAP	Lightweight Directory Access Protocol
DLL	Dynamic Link Library
XML	Extensible Markup Language
XSS	Cross-Site Scripting
ECC	Elliptic Curve Cryptography
PKRG	Public Key Random Generation

II. Prior Knowledge

We first present a unified alphabetical list of the acronyms and abbreviations used in this article (see Table I) to help readers and make the material easier to understand. This list

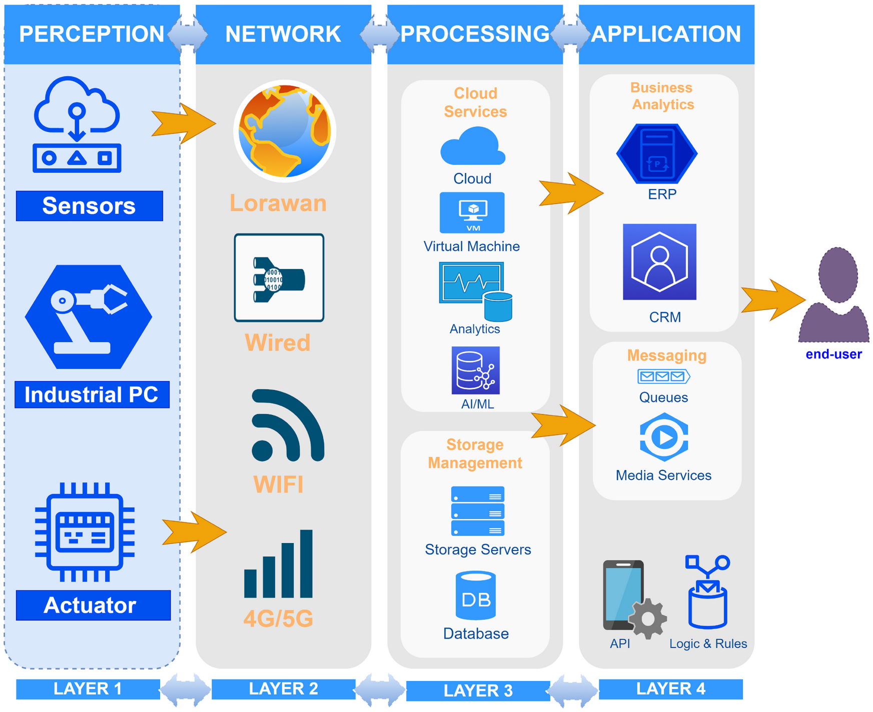


Fig. 2. General representation of the layered architecture of the IIoT system.

makes it easy for readers to search for unfamiliar acronyms and abbreviations.

A. IIoT System Architecture in General

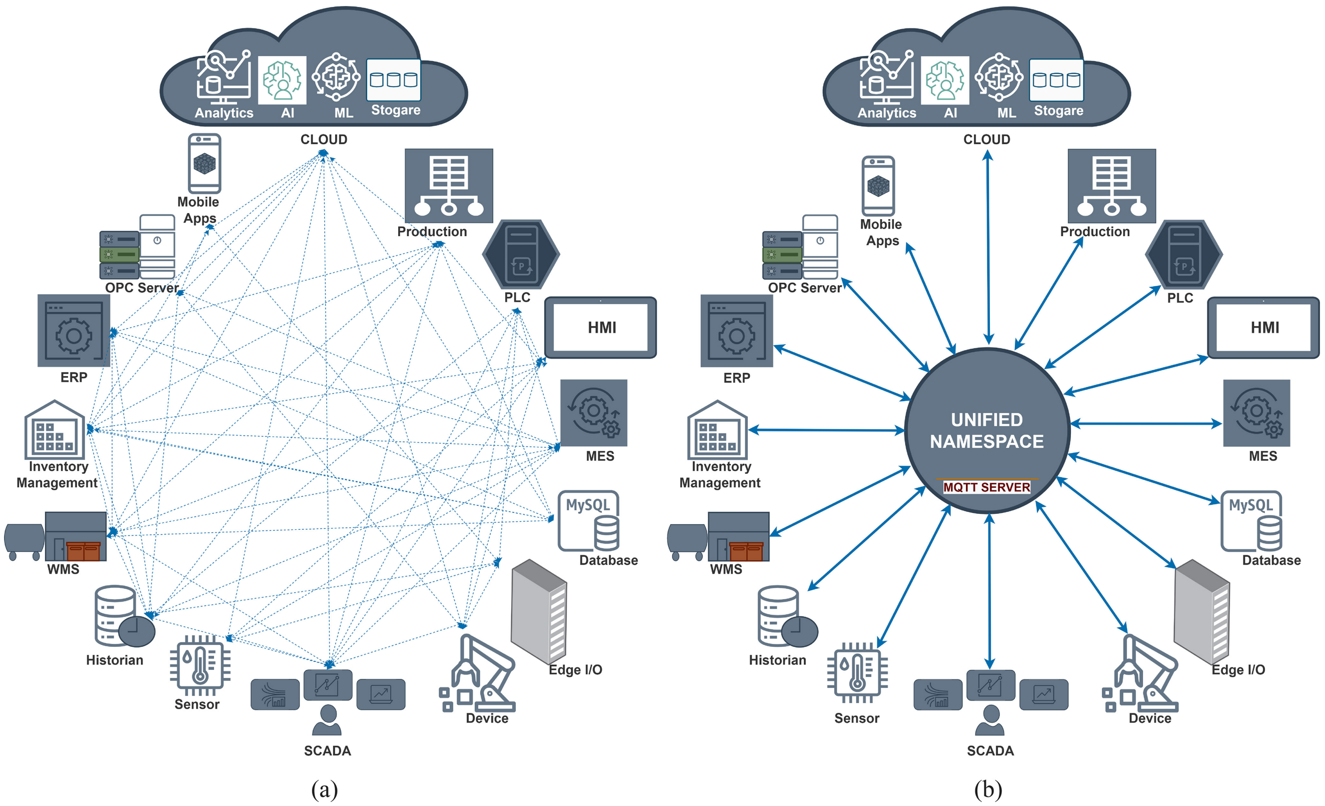
Building a typical IIoT infrastructure aims to enable rapid and reliable data access for businesses. As IIoT is new, there are no fixed guidelines for its architecture. It varies based on company demands. IIoT has four primary layers: 1) perception; 2) network; 3) processing; and 4) application. Fig. 2 shows these levels. A detailed explanation of each layer follows.

1) *Perception Layer:* At the bottom of the IIoT architectural hierarchy is the perception layer. It integrates sensors, actuators, intelligent machines, and network connectivity in the industrial unit. Sensors collect data from the physical world and send it to the cloud for analysis. Predictive models are used to control actuators, influencing industrial mechanisms.
2) *Network Layer:* The network layer is the primary information conductor throughout the IIoT system. It acts as a bridge between the perception layer, the processing layer, and the application layer. As a result, it ensures all data and control flows throughout the organization via digital sensor networks, Ethernet, or wireless technologies, such as Wi-Fi, wireless sensor networks (WSNs), WLANs, RFID, ZigBee, Bluetooth, 3G, 4G, and so on.
3) *Processing Layer:* The processing layer in cloud-based IIoT receives and analyzes data from sensors and devices. It utilizes cloud computing, including AI, machine learning (ML), and deep learning, to generate insights, facilitate predictive analysis, and improve efficiency, production, and cost in manufacturing.
4) *Application Layer:* The business applications needed for industrial organizations' growth and modernization exist in this layer. It interacts with end-node devices based on the processed data, aiding in monitoring, optimizing energy usage, setting alarms, improving uptime, controlling actuators remotely, etc.

B. Unified Namespace for IIoT Infrastructure

Before discussing why the unified namespace (UNS) is essential to properly use IIoT and Industry 4.0 principles and digitally transform a business, we first outline in more detail the functionality of the many fundamental components required to construct a typical IIoT infrastructure.

1) *PLC:* "PLC" stands for programmable logic controller, which allows industrial employees to interact with and monitor tools and processes. HMIs can be tablets, integrated screens, or computer monitors, helping users manage operations, identify issues, and understand industrial processes.
2) *HMI:* "HMI" stands for human–machine interface, enabling factory employees to interact with and monitor tools, processes, or technology. HMIs can be tablets, integrated screens, or computer monitors, essential for controlling operations and understanding industrial processes.
3) *SCADA:* A SCADA system is software that manages, tracks, and evaluates industrial processes. It communicates with field controllers, gathers real-time data, and displays it to operators via an HMI, enabling them to monitor and control the process efficiently.



(a)

(b)

Fig. 3. Typical architecture for IIoT systems. (a) IIoT systems with multiple discrete connections between nodes that do not use a UNS. (b) IIoT systems that make use of a UNS.

4) *MES:* MES are electronic programs that improve industrial productivity and reduce paper usage. They capture real-time data to accelerate production cycles. The core capabilities include work orders, scheduling, manufacturing productivity measurement, and downtime monitoring.
5) *ERP:* ERP is used by organizations to manage various tasks, such as ordering, finances, HR, and operations. It centralizes data, providing end-to-end connectivity, improving customer service, and empowering employees with real-time data for enhanced productivity.
6) *Warehouse Management System (WMS):* A WMS is a piece of software that manages all of the operations in a warehouse. WMS gives visibility and regulates crucial processes like inventory management, location management, receiving and put-away, picking, packing, and sorting. WMS removes reliance on warehouse staff for operational and inventory-related decisions.

In the vast majority of businesses today, data would either be sent from PLC to SCADA or from PLC to HMI, which would subsequently be passed to the SCADA system. There is no direct communication between the ERP system and the SCADA system, so if a business ever wanted to get information from its ERP system into the SCADA system, it would have to do two things: first, connect the ERP system to the MES system to obtain any new data the SCADA system requested, and second, map it from the MES system to the SCADA system. As a result, as corporations expand, their systems wind up having thousands of connections between different nodes. Fig. 3(a) depicts the different discrete connections that exist between the various nodes of the organization's IIoT platform.

To harness IIoT and Industry 4.0 concepts for digital transformation, we need to connect all applications seamlessly using a UNS (UNS). Each component in the plant becomes a system node. UNS acts as the information source for SCADA systems and HMIs, while PLC, MES, ERP, WMS, and other applications publish and retrieve data from it. The IIoT architecture with a UNS is depicted in Fig. 3(b). This architecture empowers real-time monitoring, analysis, and predictions, enhancing industrial efficiency and decision making. However, many businesses struggle with ML and AI projects due to the absence of a shared UNS, hindering progress and optimization across the organization's digital infrastructure.

C. Analysis of MQTT for IIoT Systems

1) Components and Structure of MQTT: The MQTT [17] is an efficient messaging protocol tailored for the IIoT ecosystem, offering scalability and wide-ranging application support. MQTT's architecture revolves around the concept of publishers and subscribers, ensuring low latency, reduced bandwidth consumption, and power efficiency. The protocol dynamically employs three levels of Quality of Service (QoS) for message delivery, allowing flexible reliability options. Its structure comprises publishers, brokers, and subscribers. Publishers act as senders, connecting to an MQTT broker over TCP for message transmission. The broker serves as an intermediary, handling both subscribing and publishing tasks, receiving messages from publishers and disseminating them to relevant subscribers. Communication within MQTT revolves around topics, serving as paths for message transmission. This structure enables a streamlined, efficient data exchange mechanism.

2) MQTT in IIoT Systems: Fig. 3(b) illustrates an example of IIoT using MQTT at its core. The MQTT protocol enables seamless communication and data exchange between the SCADA, ERP, and EMS systems, allowing them to collect and access real-time information about the state of our business. Through MQTT, these systems can efficiently share data and insights, ensuring a synchronized flow of information across the entire IIoT ecosystem. By utilizing a UNS, the data collected from these systems is stored in a standardized and organized manner, facilitating easy retrieval and analysis. This unified approach ensures that the SCADA, ERP, and EMS systems can collaborate effectively, providing a comprehensive view of our business operations and empowering informed decision making based on the up-to-date and accurate data.

On contrary, Fig. 3(a) illustrates an example of IIoT not utilizing MQTT at its core. Any IIoT architecture employing the model depicted in Fig. 3(a) is bound to face significant challenges and may ultimately fail to meet the necessary criteria for successful deployment. For instance, without the MQTT protocol, the SCADA, ERP, and EMS systems would face significant communication challenges, hindering their ability to collect and access real-time information about the state of our business. Without MQTT, the systems would lack a standardized and efficient means of transferring data, resulting in a disjointed and fragmented information flow. Consequently, the absence of MQTT would impede the creation of a UNS for storing data, making it difficult to organize and retrieve relevant information efficiently. As a result, the SCADA, ERP, and EMS systems would operate in isolation, leading to suboptimal decision making based on outdated and incomplete data, ultimately hindering our business's ability to thrive in a competitive and dynamic industrial landscape.

D. Real-World Scenarios Demonstrating the Benefits and Security Challenges of IIoT in Industrial Settings

IIoT technologies offer numerous benefits, as we observe an increasing number of companies aiming to incorporate them into their strategies. However, despite these benefits, several problems occur, particularly security and privacy issues.

1) Real-World Illustrations Showing the Advantages of IIoT Implementation: The IIoT offers various noteworthy benefits across industrial settings. The ability to increase production and efficiency through intelligent and remote supervision is one of the main factors driving its adoption by many industry Sectors. By utilizing real-time data and insights, IIoT helps prevent downtime due to equipment failures and other performance issues, which improves productivity in the workplace.

A well-known Japanese company in the automated industry, the Hirotec Group, is a prime example of the advantages of IIoT. They decreased downtime and increased production efficiency by adopting an IIoT platform. The IIoT platform provided real-time monitoring, optimized production, and guaranteed better working conditions for staff by utilizing a wide range of data sources, including robots, cameras, and sensors. At Hirotec, automated predictive maintenance shows how IIoT improves industrial operations by further increasing safety and efficiency [18], [19], [20].

Another example of the benefits offered by IIoT can be seen in smart warehousing in logistics. *DHL* and *Alibaba*, two top players in the logistics and e-commerce industries, have improved their warehouse and logistics operations by utilizing IIoT systems. DHL has embraced several technologies, including sensors in mailboxes to notify drivers of effective product pickups and IIoT installation in their facilities for better shipment tracking and storage. Aiming to use robotics in material handling, warehouses, and last-mile deliveries, they also investigate robotics and autonomous aerial vehicles to alleviate potential labor shortages in the future. In a similar vein, Alibaba automates the storage to packaging operations in smart warehouses using robotic devices outfitted with Wi-Fi signals and laser sensors, significantly lowering the need for labor and boosting productivity. These IIoT installations have led to higher process throughput, reduced labor reliance, and increased operational efficiency for both companies. These businesses have established themselves as market leaders in logistics and e-commerce by embracing IIoT technical breakthroughs and Industry 4.0 ideas [21].

2) Instances of Real-World Attacks Directed Against IIoT Systems: To substantiate our argument that IIoT systems are vulnerable to various security concerns, we evaluated some of the most relevant real-world attack scenarios that have occurred in diverse industrial sectors.

1) *Bombardier Cybersecurity Breach:* Bombardier, a Canadian manufacturer of business jets, experienced unauthorized access and data extraction by exploiting a vulnerability in a third-party file-transfer application. This application was running on purpose-built servers, isolated from the main Bombardier IT network. The company promptly responded to the incident by initiating its cybersecurity protocols and seeking the expertise of cybersecurity and forensic professionals to assess the extent of the breach. Forensic analysis revealed that personal and confidential information of employees, customers, and suppliers was compromised, affecting around 130 employees in Costa Rica [22].
2) *WannaCry Ransomware Attacks:* The NHS faced a global cyberattack on 12 May 2017, involving the WannaCry ransomware. The attack also affected other organizations, such as Telefnica, Renault, and FedEx. Although the NHS was not a direct target, its complex environment, including computing, medical, and political factors, contributed to its vulnerability. Legacy systems, such as Windows XP, were not the main cause, as WannaCry crashed computers before encryption. A Microsoft patch released two months earlier could have prevented the attack, but organizational challenges hindered its implementation across all NHS environments [23].

3) *TRITON Malware Attack:* The Triton attack on petrochemical facilities in the Middle East in 2017 raised serious concerns about ICS malware threats. Designed to target safety instrumented systems (SISs), the Triton malware aimed to disrupt or damage industrial processes, posing potential risks to human life. The sophisticated malware-embedded PowerPC shellcode and the proprietary communication protocol TriStation, granting attackers complete control over the target system. While the attack did not fully succeed, investigators warned of the catastrophic consequences had the final payload been delivered. The Triton attack represented a significant shift in ICS targeting, emphasizing the need for robust security measures to safeguard the critical infrastructure from potential cyber–physical risks [24], [25].

E. Empirical Investigations of Companies Implementing the Industrial Internet of Things in Their Operational Frameworks

As was already said, there have been significant improvements in using IIoT systems to enhance performance and efficiency in industries. Various technology-driven sectors like manufacturing, IT industries, automotive, healthcare, agriculture, and retail have started adopting IIoT technology to transform their businesses digitally [26]. This adoption allows them to make decisions faster and improve their operations. In the following, we will explore real-life examples of companies that have successfully included IIoT in their operational processes.

1) Case Study 1—The Dutch SME 247 TailorSteel: 247TailorSteel, an SME, effectively implements IIoT in its fully automated factory [27]. With 125 employees, a 30 million euro turnover in 2014, and 25%–30% annual growth, it exemplifies Industry 4.0. Their proprietary software, SOPHIA, automates production, logistics, and online orders, reducing costs and ensuring faster operations. Robotic vehicles oversee production, ensuring efficiency. The customer-centric approach allows small-batch orders without cost variations. Expansion plans include establishing additional European factories to enhance accessibility and customer satisfaction.

2) Case Study 2—Siemens: Siemens is a remarkable example of a business embracing IIoT, with its platform brand, MindSphere, introduced in 2016 as a commercial IIoT service catering not only to manufacturing customers but also extending its reach beyond, forming the core of the company's IIoT strategy [28]. MindSphere offers various features, such as machine connectivity and front-end IIoT apps. Initially relying on SAP's infrastructure services, Siemens later established robust partnerships with leading cloud service providers like Amazon and Microsoft to solidify its position within the complex IIoT landscape. Leveraging its technological expertise in industrial networks and controllers at the connectivity layer, Siemens gains a critical advantage in the IIoT ecosystem, as emphasized by multiple interviewees. Customers who have implemented Siemens' factory automation systems often opt for MindSphere as their preferred platform solution. Additionally, MindSphere integrates diverse new connectivity solutions, including open standards, facilitating seamless connections with various industrial devices. Siemens has successfully cultivated a portfolio of applications connected with MindSphere, significantly enhancing the commercial appeal of IIoT systems. Many of these applications were developed and introduced by Siemens or its affiliated companies. Siemens also actively encourages third-party development, exemplified by its 2018 acquisition of Mendix, an American software manufacturer specializing in user-friendly "low-code programming" solutions for application developers, a strategic initiative reinforcing Siemens' commitment to a thriving IIoT ecosystem.

F. Limitations of IIoT System Architecture

Despite the advantages, the IIoT system architecture has limitations that require attention to fully leverage its capabilities in industrial ecosystems. Below, we critically examine these limitations across the four primary layers, considering their impacts on reliability, scalability, interoperability, and security.

1) The heterogeneity of IoT devices presents reliability challenges with limitations in hardware and software components, affecting parameters like throughput and data accuracy [29], [30]. Sensors and actuators in remote IIoT habitats face environmental factors, such as extreme temperatures and mechanical wear, leading to hardware failures and malfunctions [31]. An example of such challenges is the Boeing 787 Dreamliner, which experiencing unexpected shutdowns due to a software glitch in its IIoT-enabled electrical system, resulting in performance issues [32].
2) Moreover, the reliance on devices with limited computational capabilities, memory storage, and battery power in IIoT systems introduces additional challenges to reliability. These constraints can result in energy consumption and time delay issues during computation offloading, further exacerbating reliability concerns [33].
3) Moreover, the reliance on devices with limited computational capabilities, memory storage, and battery power in IIoT systems introduces additional challenges to reliability. These constraints can result in energy consumption and time delay issues during computation offloading, further exacerbating reliability concerns [33].
4) Stable network connectivity and resilient communication methods are crucial for IIoT system reliability. However, the architecture itself poses difficulties in terms of communication protocol inefficiencies, network congestion, and latency problems [34], [35], [36]. With more devices connected, network congestion can happen, which can cause packet loss, increased latency, and delays in data transmission [34]. When mobile-edge computing (MEC) is used, it makes network congestion worse, especially for tasks requiring a lot of data bits, which could result in task failure and a long queue [35]. The present methods for time-slotted packet scheduling, which are frequently used in IIoT systems to achieve

the appropriate QoS, are hampered by network overhead and scalability issues [36]. Furthermore, using heterogeneous communication protocols for data transfer creates communication gaps between devices in the field and the cloud, increasing latency and decreasing reliability [37].

5) Within the processing layer of IIoT systems, data loss or inconsistent data is a reliability risk. Intermittent or unreliable network connections can cause data packets to be lost or delayed. The heterogeneity of devices and data formats conveyed via IIoT networks raises the possibility of data parsing mistakes in the processing layer, which could result in inaccurate processing results [38]. The increased volume of data could overwhelm the processing layer, causing bottlenecks that hinder real-time analysis and response and lower system reliability [39]. Insufficient error-handling mechanisms within the processing layer may compromise reliability since errors during data processing may not be effectively captured or recovered.
6) The application layer of IIoT systems, while not subject to the same constraints as the network or device layers, relies on the reliability of the lower layers. Inadequate data from the lower layers, such as inconsistencies, anomalies, or a lack of integrity, can compromise the reliability of the application layer. Application failures triggered by inadequate input validation or inefficient resource management within the application layer can render the entire application unreliable, impacting overall system reliability. Insufficient anomaly detection techniques and data integrity checks further exacerbate reliability concerns [38].
7) The IIoT system architecture faces difficulties with scalability in several areas. Scalability problems are due to the IIoT systems' exponential device expansion, diversity of networks, heterogeneity, and enormous volume of generated data. Scalability problems include managing large amounts of data transfer, addressing and recognizing devices, and supplying networking capabilities. IIoT systems face additional difficulties in terms of scalability due to energy consumption and the need to replenish depleted batteries [40].
8) Security is a critical concern in IIoT systems, given the value of the generated data and the resource-constrained nature of the communication network. Traditional security mechanisms are often insufficient to protect the complexity of IIoT systems, necessitating lightweight cryptography and privacy assurance. Interoperability challenges arise from the high heterogeneity of devices, technologies, and standards, requiring solutions to ensure seamless communication and integration. Data format incompatibility and the handling of big data introduce additional challenges to interoperability [37], [41].

III. Security in IIoT Systems

This section examines in-depth several security issues relating to IIoT systems. A hacking scenario is used as a strategy at each tier of the IIoT architecture to first classify the different types of vulnerabilities that can be exploited. Then, depending on these flaws, we present which assaults the hacker can use to compromise the security on the related layer, and ultimately, we analyze all of the consequences, direct impacts, and corresponding countermeasures that follow such security attacks.

A. Hacking Scenario on Perception Layer

As previously stated, the perception layer is a physical layer that houses IIoT devices responsible for detecting and collecting data about their immediate surroundings while also managing critical infrastructure to improve industrial machinery and production. This layer's vulnerabilities can pose a significant threat to industrial organizations, leading to cybersecurity risks. IIoT devices are more susceptible to attacks due to their exposure to the outside world. Understanding the potential vulnerabilities in this layer helps us identify various types of attacks that could target it. To demonstrate this, let's consider a hypothetical automobile manufacturing plant called '‘Prime MotorX" that produces cars, trucks, and other motor vehicles as a case study. In our hacking scenario, we explore the weaknesses in the IIoT devices and sensors used in the Prime MotorX. We discuss how adversaries exploit these vulnerabilities to perform their attacks and describe potential attacks and their effects. Moreover, we provide security measures to prevent such attacks from spreading within Premier MotorX's system, as shown in Fig. 4.

1) Security Concerns in Prime MotorX's Perception Layer: We will explore the potential vulnerabilities in the Prime MotorX perception layer that a malicious attacker could exploit to attack a specific target within Prime MotorX. The term "target" refers to the entity, system, device, or resource that the adversary aims to compromise or gain unauthorized access. These targets include but are not limited to IIoT devices, user credentials, servers, software, network infrastructure, critical infrastructure, or even the entire organization.

Scenario−1: Prime MotorX is equipped with hydraulic presses used for molding and shaping various vehicle components. The company utilizes a PLC to regulate the pressure and timing of the presses, ensuring precise and consistent formation of parts according to required specifications. However, there is a vulnerability risk as the PLC is widely exposed in the field with inadequate oversight and unlimited access, making it easily accessible to unauthorized individuals. The poor hardening of the PLC makes it possible for malicious attackers to gain access to the PLC, thus allowing them to physically damage or tamper with it. The effects of damaging the PLC can disrupt manufacturing. The presses may malfunction, leading to faulty vehicle parts and production stoppage until repaired, costing Prime MotorX valuable time and money. The effects of tampering with the PLC can have different outcomes. Wrong adjustments may lead to poorly formed vehicle parts, impacting product quality. Severe tampering can cause the presses to malfunction, causing production delays and equipment damage. Unauthorized access can also pose security risks, with control over critical processes leading to safety hazards or data

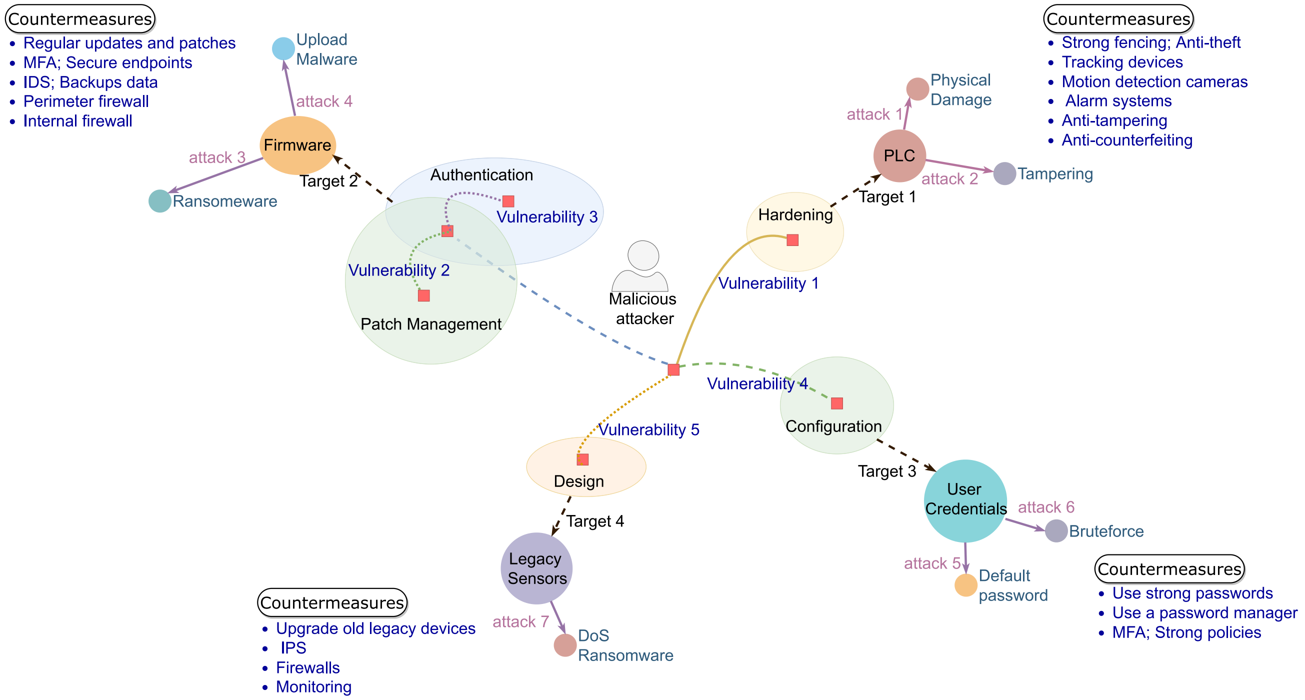


Fig. 4. Hacking scenario on the IIoT perception layer is utilized here to demonstrate the frequent vulnerabilities encountered in this layer, possible attacks arising from these flaws, and countermeasures to be put in place against these attacks.

breaches. In this scenario, the exploited vulnerability is Poor Hardening and the launched attacks are Physical Damage [42] and Tampering [43], with the target being the PLC.

Scenario—2: Prime MotorX is equipped with thermocouples to maintain precise and consistent temperature conditions during various stages of its manufacturing process. These thermocouples ensure that critical components and materials operate within the required temperature ranges, which is crucial for producing high-quality vehicles. The thermocouples have firmware software that enables them to carry out particular tasks, like precisely sensing temperature and sending data to other devices or systems inside the production facility. However, the firmware software has not been updated with the latest security patches since last month, and as a result, it is unable to address known vulnerabilities or weaknesses. Moreover, Prime MotorX has neglected to enable authentication checks mechanisms on the firmware software of all its thermocouples. By not implementing these authentication measures, the thermocouples cannot verify the legitimacy of any updates or commands they receive and, thus, cannot prevent unauthorized access to the firmware. A malicious actor may exploit these vulnerabilities to launch a ransomware and malware attack on the thermocouples' firmware (for example, by sending malware or harmful software when the firmware requests an update from the server). The effects of such an attack on the outdated firmware can disrupt the proper functioning and operations of the thermocouples, compromise data integrity, disrupt operations, and potentially lead to a complete shutdown of Prime MotorX's industrial system. Similarly, launching malware on the weak authenticated firmware can cause the device to malfunction, infect other devices, lead to critical data loss, and put the company's business at risk. In this scenario, the exploited vulnerabilities are Patch Management Failure and Weak Authentication. The launched attacks include Ransomware [44], [45], and Malware [46], with the target being the Firmware.

Scenario—3: Prime MotorX is equipped with resistance temperature detectors (RTDs) that have similar capabilities to thermocouples in monitoring temperature. RTDs are highly accurate and reliable, providing precise temperature readings in a wide range of temperatures. These RTDs are connected to PLCs with firmware software that interprets and utilizes the temperature data for various operational purposes. The firmware requires authentication checks, such as passwords, for logging in. However, some employees continue to use the default preconfigured password provided by the RTDs' supplier, while others use weak passwords. A malicious actor could exploit these vulnerabilities to launch default passwords and brute force attacks on user credentials. For those using the standard password, the attacker could search for press releases revealing the default administrator password, gain unauthorized access to private information, damage devices, and launch Denial-of-Service (DoS) attacks. For users with weak passwords, the attacker could use code to guess probable combinations to obtain the real password and seize control of the RTDs, potentially compromising production data and other devices. In this scenario, the exploited vulnerabilities are weak Authentication and Misconfiguration. The launched attacks include default password attack [47], [48], DoS attack [49], [50], and Brute force attack [51], [52], with the target being the user credentials.

Scenario—4: A wireless sensor is built into Prime MotorX to help the manufacturing assembly robot. The sensor is prone to manipulation because it is an older model with inadequate processing power. A ransomware assault could be launched on the sensor by an attacker, with serious repercussions. The obsolete sensor might not have the necessary security updates, leaving it open to ransomware infection and data encryption. As a result, the sensor's performance might be compromised, which could result in the loss of crucial data and have an effect on the entire production process. Production delays, financial losses, and significant hazards to the integrity and safety of the manufacturing plant could all result from this attack.In this scenario, the exploited vulnerability is Design, and the launched attack is a Ransomware [53], [54] on the Legacy Sensor.

2) Countermeasures for Security Issues in the Perception Layer: Based on the analysis of the different vulnerabilities and security attacks in the perception layer previously listed in scenarios 1, 2, 3, and 4, we discuss for each case study the various security measures that must be put in place to ensure the protection and security of IIoT devices.

Countermeasures for Scenario−1: We have discussed the occurrence of Physical Damage and Tampering attacks targeting the perception layer of the IIoT in our hacking scenario-1. In this section, we will explore and examine diverse countermeasures that can be effectively employed to mitigate the impact of these attacks.

Physical Damage Attacks−Countermeasures: Strong fences can be installed around critical equipment and infrastructure to prevent unauthorized physical access. Additionally, the use of anti-theft and tracking devices can deter potential attackers and facilitate the recovery of stolen assets. Furthermore, the deployment of motion detection cameras and alarm systems can enhance security measures, providing real-time alerts in case of suspicious activities. Regular inspection and assessment of equipment can also help in identifying any signs of tampering or damage at an early stage, enabling prompt corrective actions. In addition to the above measures, the adoption of anti-tampering and anti-counterfeiting technologies can further safeguard the integrity of the perception layer. These technologies can help detect any attempts to tamper with or counterfeit components and equipment, ensuring the authenticity and reliability of the IIoT devices. By employing these comprehensive countermeasures, organizations can effectively defend against Physical Damage attacks and enhance the overall security posture of their IIoT infrastructure [55], [56], [57].

Tampering Attacks−Countermeasures: The adoption of anti-tampering and anti-counterfeiting technologies can further safeguard the integrity of the perception layer. These technologies can help detect any attempts to tamper with or counterfeit components and equipment, ensuring the authenticity and reliability of the IIoT devices.Implement secure boot and code signing to prevent unauthorised code change, use cryptographic techniques to preserve data integrity, making it harder for attackers to modify data during transit or storage. By employing these comprehensive countermeasures, organizations can effectively defend against Tampering attacks and enhance the overall security posture of their IIoT infrastructure [57].

Countermeasures for Scenario−2: We have discussed the occurrence of Physical Ransomware and Malware attacks targeting the perception layer of the IIoT in our hacking scenario-2. In this section, we will explore and examine diverse countermeasures that can be effectively employed to mitigate the impact of these attacks.

Ransomware Attacks−Countermeasures: To defend against ransomware attacks in the IIoT perception layer, implementing cyber threat hunting (CTH) helps proactively identify and mitigate potential threats before they escalate. Regular firmware and device patches address known vulnerabilities, minimizing the attack surface for ransomware. Furthermore, introducing more robust and secure endpoints enhances device security, reducing the likelihood of successful ransomware infiltrations. By adopting these countermeasures, the IIoT perception layer can bolster its resilience against ransomware attacks and safeguard critical industrial operations and data from potential harm [58], [59], [60].

Malware−Countermeasures: To defend against malware attacks in the IIoT perception layer, the implementation of sophisticated cybersecurity frameworks offers comprehensive protection and threat detection capabilities. Employing dependable authentication methods, like multifactor authentication (MFA), during updates ensures that only authorized personnel can access and modify critical components, reducing the risk of malware infiltration. Additionally, leveraging the interplanetary file system (IPFS) for data storage and retrieval enhances data integrity and availability, mitigating the impact of potential malware-induced data loss. Creating regular backups of data on devices adds an extra layer of defense, safeguarding against data loss or corruption in the event of a malware attack. These countermeasures fortify the IIoT perception layer, bolstering its resilience against malware and safeguarding the integrity and availability of essential industrial processes and data [61], [62], [63], [64].

Countermeasures for Scenario−3: We have discussed the occurrence of Default Password attacks and Brute Force attacks targeting the perception layer of the IIoT in our hacking scenario-3. In this section, we will explore and examine diverse countermeasures that can be effectively employed to mitigate the impact of these attacks.

Default Password−Countermeasures: Countermeasures against Default Password Attacks in the IIoT perception layer involve eliminating the use of default passwords and encouraging the implementation of strong, unique passwords for all devices and accounts. Manufacturers and users should ensure that default credentials are changed upon initial setup to prevent unauthorized access. Implementing a one time password (OTP) system adds an extra layer of security, generating time-limited and unique passwords for each login attempt. Password managers can help users maintain complex and diverse passwords without the risk of forgetting or reusing them. Additionally, employing MFA reinforces the defense against default password attacks, requiring users to provide multiple forms of identification for access, making it significantly harder for malicious actors to gain unauthorized entry. Moreover, ensuring secure endpoints with up-to-date firmware and robust security measures can prevent the exploitation of devices with default passwords, bolstering the overall resilience of the IIoT perception layer [65], [66], [67].

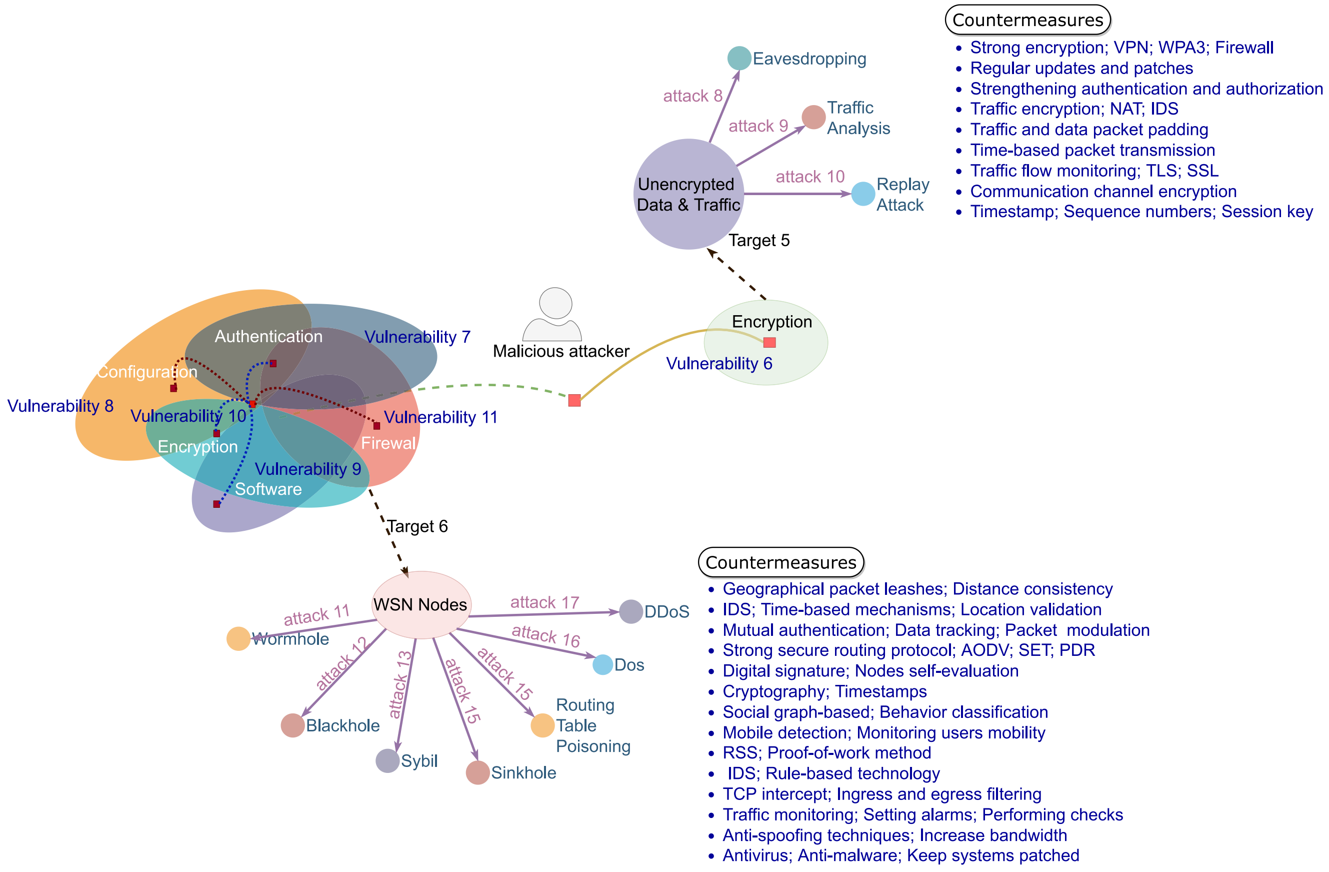

Countermeasures
• Strong encryption; VPN; WPA3; Firewall
• Regular updates and patches
• Strengthening authentication and authorization
• Traffic encryption; NAT; IDS
• Traffic and data packet padding
• Time-based packet transmission
• Traffic flow monitoring; TLS; SSL
• Communication channel encryption
• Timestamp; Sequence numbers; Session key
Eavesdropping
attack 8
attack 9
Traffic Analysis
attack 10
Replay Attack
Unencrypted Data & Traffic
Target 5
Encryption
Vulnerability 6
Malicious attacker
Authentication
Vulnerability 7
Configuration
Vulnerability 8
Vulnerability 10
Vulnerability 11
Firewal
Encryption
Vulnerability 9
Software
Target 6
Countermeasures
• Geographical packet leashes; Distance consistency
• IDS; Time-based mechanisms; Location validation
• Mutual authentication; Data tracking; Packet modulation
• Strong secure routing protocol; AODV; SET; PDR
• Digital signature; Nodes self-evaluation
• Cryptography; Timestamps
• Social graph-based; Behavior classification
• Mobile detection; Monitoring users mobility
• RSS; Proof-of-work method
• IDS; Rule-based technology
• TCP intercept; Ingress and egress filtering
• Traffic monitoring; Setting alarms; Performing checks
• Anti-spoofing techniques; Increase bandwidth
• Antivirus; Anti-malware; Keep systems patched
WSN Nodes
attack 11
Wormhole
attack 17
DDoS
attack 16
Dos
attack 12
Blackhole
attack 13
Sybil
attack 15
Sinkhole
attack 15
Routing Table Poisoning

Fig. 5. Common vulnerabilities found in this layer, the attacks arising from these vulnerabilities, and the countermeasures to be implemented against these assaults are all described, accordingly, using a hacking scenario on the network layer of IIoT.

Brute Force—Countermeasures: Countermeasures against Brute Force Attacks in the IIoT perception layer include setting account lockout policies that temporarily lock accounts after a specified number of failed login attempts, hindering brute force attackers from making repeated guesses. Utilizing strong passwords, as mentioned previously, further strengthens defense by increasing the complexity of potential combinations, rendering brute-force attacks less feasible. Enforcement of laws and regulations on device management can also mandate secure authentication practices, discouraging manufacturers and users from employing weak password policies. Continuous monitoring of login attempts and suspicious activities can help detect and prevent brute force attacks in real time, allowing for timely response and mitigation. Combining these countermeasures reinforces the security posture of the IIoT perception layer, safeguarding against brute force attacks and ensuring the protection of critical industrial processes and data [66], [67].

Countermeasures for Scenario—4: We have discussed the occurrence of Physical Ransomware attacks targeting the old legacy wireless sensor devices on The IIoT perception layer in our hacking scenario-4. In this section, we will explore and examine diverse countermeasures that can be effectively employed to mitigate the impact of these attacks.

Ransomware on Old Devices—Countermeasures: To defend against Ransomware attacks targeting old legacy wireless sensor devices in the IIoT perception layer, one effective countermeasure is to address design vulnerabilities by replacing these outdated devices with more secure and modern alternatives. Upgrading to newer wireless sensor devices ensures that they are equipped with the latest security features, making them less susceptible to Ransomware attacks. Legacy devices may lack robust security mechanisms, making them easy targets for malicious actors. By eliminating these old legacy devices and adopting newer ones with enhanced security protocols, organizations can better protect their IIoT infrastructure from Ransomware threats and mitigate potential risks associated with outdated technology [68].

B. Hacking Scenario on Network Layer

To map the vulnerabilities, security threats that exploit those flaws, and different measures to take against these attacks (see Fig. 5), we use the same analogy as in the previous hacking scenario.

1) Security Concerns in Prime MotorX's Network Layer: We will explore the potential vulnerabilities in the Prime MotorX network layer that a malicious attacker could exploit to attack a specific target within Prime MotorX.

Scenario−5: In this situation, a malicious actor knows certain sensors at Prime MotorX that send data to the cloud for analysis through radio communication, but these sensors lack encryption for their data. This vulnerability allows the attacker to perform an eavesdropping attack, intercepting the unencrypted data and accessing critical information about Prime MotorX. To address this issue, Prime MotorX's IT security team decided to encrypt the sensor data before sending it to the cloud, making it unreadable to the attacker. However, the attacker can still capture the encrypted data and carry out a Traffic Analysis attack using tools like Wireshark to monitor the connection between the sensor and the cloud and access the data based on their observations. Additionally, the attacker may employ a replay attack, repeatedly sending the captured data to the cloud server to cause confusion and unauthorized effects. In this scenario, when the data is unencrypted, the attacker exploits insufficient data and transport encryption as a vulnerability and launches an Eavesdropping [69], [70], with the target being Unencrypted Data. On the other hand, when the data is encrypted, the attacker launches a Traffic Analysis attack [71], utilizing Wireshark [72], and a Replay attack [73], with the target being the encrypted data. Eavesdropping and Traffic Analysis are passive attacks, whereas the Replay attack is an active attack.

Scenario−6: In this case, a malicious actor aims to launch a series of attacks on Prime MotorX's network. Since direct access to the network is not possible, the adversary decides to explore security vulnerabilities in the IIoT end devices and system programs of Prime MotorX. These vulnerabilities include misconfigured authentication mechanisms, default configuration, outdated software, weak encryption, and firewall issues. By exploiting these weaknesses, the adversary seeks to obtain user credentials or bypass authentication to take control of the nodes. This would allow him to gain access to the network and carry out malicious activities. The target of the attacks is two nodes of Prime MotorX's WSN. The malicious actor can choose to launch various attacks, including the Wormhole attack without prior knowledge of the network. This attack involves using a compromised nodeX close to a legitimate node in the network to capture packets from nodeX and tunnel them to another compromised nodeY at the other end of the network. The packets are then forwarded to the legitimate receiving node and its neighboring nodes, causing delays and disruption in the routing algorithm. Another potential attack is the Blackhole attack, where the compromised nodeX deceives the source code into believing it has the shortest path to the destination node. However, nodeX drops all packets sent to nodeB, resulting in data loss and potential financial consequences for the company Prime MotorX. The attacker may also consider the Sybil attack by assigning multiple fake identities or solids to malicious nodes. These fake identities could be used to gain unauthorized access to the network and manipulate the transmission flow of data packets, compromising data integrity and network stability. The Sinkhole attack is another option, where the malicious actor tricks neighboring nodes into believing that nodes have the shortest path to a base station (BS) serving as a communication link between sensor network nodes. This allows him to direct all traffic from neighboring nodes to itself, enabling various attacks on the security and integrity of Prime MotorX. The attacker can employ the Routing Table Poisoning attack using the infected nodes to corrupt the routing tables of other legitimate nodes in the network. This leads to routing loops, inefficient routing, and bandwidth bottlenecks, significantly damaging the network's performance. Two types of DoS attacks can also be launched by the attacker. The first is the traditional DoS attack, where the adversary overloads a server of Prime MotorX with multiple TCP connection requests, preventing legitimate users from accessing it and causing production slowdowns and financial losses. The other type is the Distributed DoS (DDoS) attack, which involves using multiple computers and network connectivity (botnet) to launch a coordinated attack on the target. This amplifies the impact, causing severe disruptions, loss of revenue, and data breaches. These potential attacks demonstrate the critical need for Prime MotorX to address the identified vulnerabilities promptly and implement robust security measures to safeguard its IIoT infrastructure and prevent potential attacks from malicious actors.In this scenario, the exploited vulnerabilities encompass Weak Authentication and Misconfiguration. The launched attacks consist of the Wormhole attack [74], Blackhole [58], [75], the Sybil attack [60], the Sinkhole attack [76], the Routing Table Poisoning attack [77], DoS attack [78], [79], the DDoS attack [79], [80], with the target being the WSN Nodes of Prime MotorX.

2) Countermeasures for Security Issues in the Network Layer: In this section, we emphasize the available defenses against the numerous security threats to the IIoT network layer that were previously described in scenarios 5 and 6.

Countermeasures for Scenario−5: We have discussed the occurrence of Eavesdropping, Traffic Analysis, and Replay attacks targeting the network layer of the IIoT in our hacking scenario 5. In this section, we will explore and examine diverse countermeasures that can be effectively employed to mitigate the impact of these attacks.

Eavesdropping−Countermeasures: To defend against eavesdropping attacks on the IIoT network layer, implementing robust preventative measures is essential. Strengthening wireless connections through enhanced authentication and authorization processes, such as adopting WPA3, can significantly reduce the risk of unauthorized access. Installing firewalls and regularly updating software helps to fortify the network's defenses against potential vulnerabilities. Additionally, using a virtual private network (VPN) when connecting to public Wi-Fi can further secure data transmission and prevent potential eavesdroppers from intercepting sensitive information. By proactively incorporating these countermeasures, organizations can bolster the security of their IIoT network layer and safeguard against eavesdropping threats [81], [82], [83].

Traffic Analysis Attacks−Countermeasures: To defend against traffic analysis attacks on the IIoT network layer, several countermeasures can be implemented. Employing strong encryption algorithms to encrypt both internal and external traffic enhances data security and confidentiality, making it difficult for attackers to decipher the information being transmitted. Utilizing network address translation (NAT) to redirect traffic on IIoT devices adds an additional layer of complexity, making it challenging for attackers to discern the nature of the communication. Controlling the timing of packet transmission and adopting differential privacy techniques further obfuscates traffic patterns, preventing potential attackers from extracting meaningful insights. Implementing real-time traffic flow analysis and detection mechanisms helps identify and thwart malicious activities, enhancing the overall security of the IIoT network layer. By combining these countermeasures, organizations can effectively safeguard against traffic analysis attacks and protect the integrity and privacy of their data and communications [84], [85], [86].

Replay Attacks—Countermeasures: For security countermeasures against replay attacks on the IIoT network layer, several measures can be implemented. First, ensuring the encryption of the communication channel between nodes and the client-server using a secure protocol like TLS or SSL adds a layer of protection against intercepted and replayed data. Additionally, applying timestamps and sequence numbers on each transmitted data packet helps to prevent replay attacks by enabling the detection of duplicate or outdated packets. Alternatively, employing a session key during transactions between communicated nodes ensures that the generated key can only be used for a specific session and becomes invalid for subsequent transactions, thereby thwarting replay attempts. By incorporating these countermeasures, organizations can enhance the security of their IIoT network layer and defend against replay attacks effectively [87], [88], [89], [90].

Countermeasures for Scenario—6: We have discussed the occurrence of Wormhole, Blackhole, Sybil,Sinkhole, Routing Table Poisoning, and DoS/DDoS attacks targeting the network layer of the IIoT in our hacking scenario-6. In this section, we will explore and examine diverse countermeasures that can be effectively employed to mitigate the impact of these attacks.

Wormhole Attacks—Countermeasures: To counter wormhole attacks in the IIoT network layer, a multifaceted approach can be employed. Implementing detection measures, such as AODV, intrusion detection system (IDS), IPS algorithms, RPL, time-based mechanisms, location validation, geographical packet leashes method, and distance metrics like distance consistency aids in identifying and mitigating potential wormhole threats. Additionally, deploying secure route selection and mutual authentication mechanisms, along with the use of the secure neighbor discovery (SEND) protocol, helps in preventing and stopping wormhole attacks. By tracking the path of data packets and employing modulation packet techniques on wireless communication after experiencing an attack on the nodes, the impact of wormholes can be minimized, enhancing the overall security and resilience of the IIoT network layer against such attacks [91], [92], [93], [94], [95], [96].

Blackhole Attacks—Countermeasures: To defend against blackhole attacks in the IIoT network layer, robust security measures can be implemented. Deploying a strong secure routing protocol like Ad-hoc on-demand distance vector (AODV), SET (secure efficient ad hoc distance vector routing), or position-based directed routing (PDR) enables the selection of optimal paths, effectively countering malicious routes offered by attackers. Additionally, empowering nodes with the ability to self-evaluate the proportion of packets delivered by surrounding nodes and utilizing digital signatures to secure the AODV routing protocol further enhances the network's resilience against blackhole attacks. These countermeasures help ensure reliable and secure data transmission within the IIoT network, safeguarding against potential disruptions and protecting critical infrastructure and communication channels [91], [92].

Sybil/Sinkhole/Routing Attacks—Countermeasures: To defend against Sybil attacks on the IIoT network layer, a multifaceted approach can be employed. Utilizing strong cryptography is crucial to protect against Sybil, sinkhole, and routing table poisoning attacks. Incorporating timestamps in data helps ensure data integrity and authenticity. Social graph-based, PDR, behavior classification, and mobile detection techniques can aid in identifying and mitigating Sybil attacks. Monitoring users' actual mobility and utilizing received signal strength indicator (RSSI) as an indicator are effective methods to detect potential Sybil attackers. Implementing Proofs-of-Work approach can prevent spam and thwart Sybil attacks. Additionally, deploying IDSs, rule-based technologies, and geo-routing protocols further fortify the network's defense against Sybil attacks, ensuring the IIoT network remains secure and resilient to potential threats [97], [98], [99].

DoS/DDoS—Countermeasures: To defend against DoS and DDoS attacks on the IIoT network layer, a comprehensive set of security measures can be implemented. Employing ML techniques, TCP intercept, ingress and egress filtering, and monitoring with alarms for malicious activity aids in detecting and preventing such attacks. Utilizing encryption based on the TLS protocol and tokens enhances data security. Antispoofing techniques can be employed to block spoofed packets from entering or exiting the network, and regular checks should be executed to ensure no DoS and DDoS entities have infiltrated the network. Increasing bandwidth capacity enables better handling of traffic spikes caused by DoS and DDoS attacks. Regular system updates and installation of antivirus and anti-malware software further bolster the network's resilience against potential threats. By adopting these countermeasures, the IIoT network layer can effectively mitigate the impact of DoS and DDoS attacks, ensuring the smooth and secure operation of the industrial systems [100], [101], [102].

C. Hacking Scenario on Processing Layer

As shown in Fig. 6, a hacking scenario will also be used to identify the variety of cybersecurity issues found in this layer, as well as the different attack paths that malicious actors can employ to exploit these flaws and the various protections and prevention measures against these attacks.

1) Security Concerns in Prime MotorX's Processing Layer: We will explore the potential vulnerabilities in the Prime MotorX network layer that a malicious attacker could exploit to attack a specific target within Prime MotorX.

Scenario—7: In this situation, the attacker discovers that the existing security controls provided by the CSP of Prime

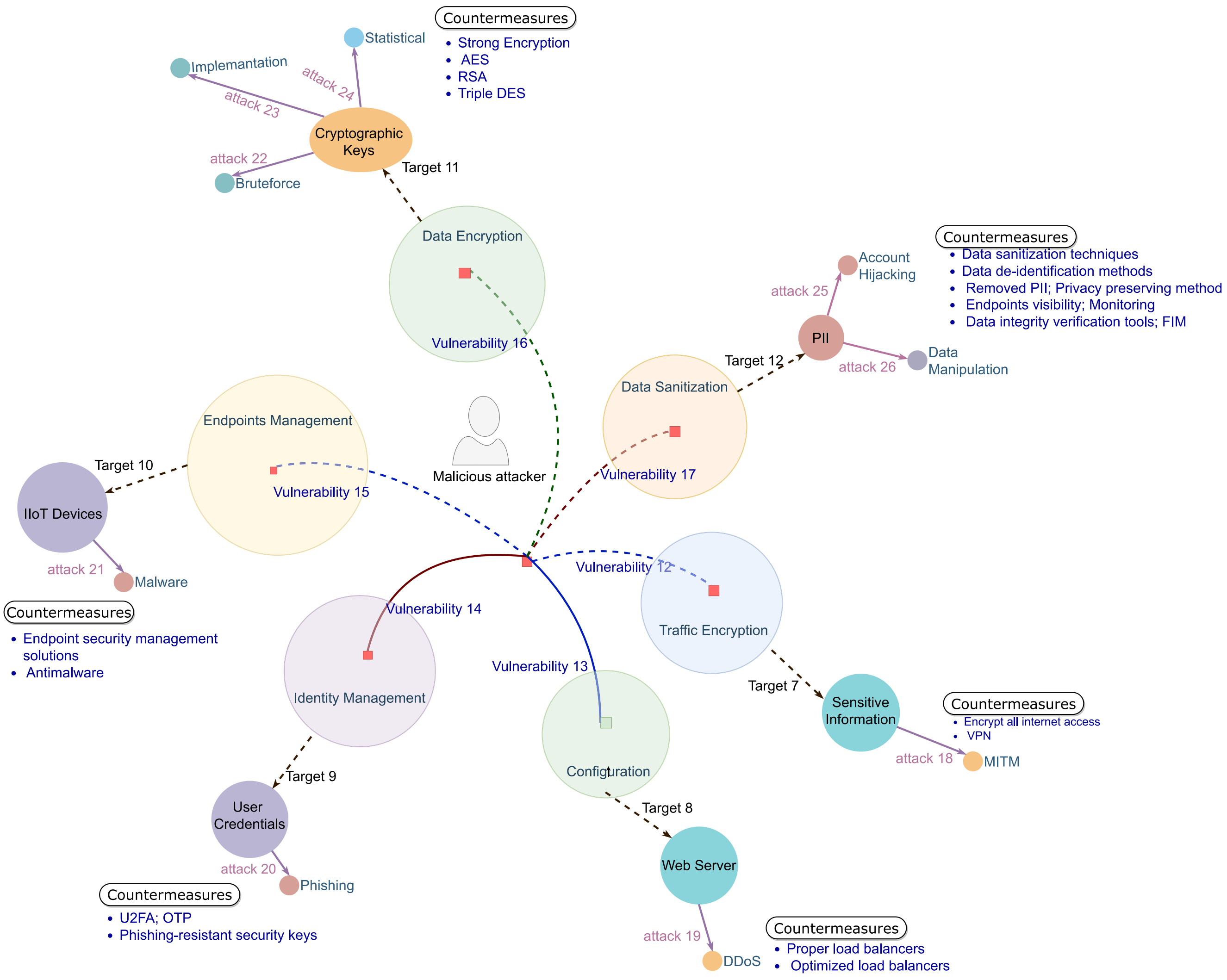


Fig. 6. Common vulnerabilities found in this layer, the attacks arising from these vulnerabilities, and the countermeasures to be implemented against these assaults are all described, accordingly, using a hacking scenario on the Processing layer of IIoT.

MotorX have not been correctly implemented, leading to vulnerabilities that grant unauthorized access to the processing layer's resources. One example is the attacker's use of a MITM attack [103] to intercept Web traffic and access sensitive information within Prime MotorX's processing layer due to the lack of Web traffic encryption. This attack can have severe consequences, enabling the attacker to obtain user credentials and sensitive corporate and customer data, potentially disrupting production or gaining control over Prime MotorX's entire IT environment. Additionally, the attacker may exploit misconfigured cloud-level load balancing, causing a DDoS attack [104] that renders Prime MotorX resources on the Web server inaccessible, disrupting production and customer experience. Moreover, the attacker may discover poorly configured identity authorization services, allowing them to launch a Phishing attack [105] to steal user credentials and access resources at the processing layer. If Prime MotorX's user devices have weak endpoint security management, the attacker can exploit this vulnerability to launch a Malware attack [106] and gain unauthorized access to the IIoT devices. Addressing these vulnerabilities is crucial for Prime MotorX to ensure the security and integrity of its IT infrastructure and protect against such potential attacks.

Scenario—8: In this particular case, the attacker has gained unauthorized access to Prime MotorX's processing layer and now aims to directly attack the stored data. To achieve this, the attacker seeks vulnerabilities in the data to compromise data privacy or manipulates the data to compromise data integrity. One approach is to exploit poor data encryption at the data center system level by employing Brute Force attack [107], Implementation attack [108], or Statistical attack [109] on weak cryptographic algorithms to crack the encryption keys and access sensitive information. The exploited vulnerability is poor data encryption, and the target is the cryptographic keys. Once the data is stolen, the attacker may also compromise the personally identifiable information (PII) of Prime MotorX employees or customers, hijacking their accounts through Account Hijacking attack [110], [111], and damaging their identity integrity due to the lack of adequate data sanitization mechanisms by the CSP. The exploited vulnerability is the lack of data sanitization, and the target is the PII of users. Such a data privacy breach can lead to severe consequences for both customers and the company, including financial losses, reputational damage, and decreased trust in the business. Moreover, the attacker may choose to manipulate data through Data Manipulation attack [112] directly at the data center system level, leading to data integrity degradation and causing the Prime MotorX system to malfunction.

2) Countermeasures for Security Issues in the Processing Layer: Subject to the study of the various flaws and assaults launched on the IIoT processing layer formerly enumerated in scenarios 7 and 8, we discuss in this section the diverse precautions to be taken against security concerns in this layer.

Countermeasures for Scenario—7: We have discussed the occurrence of MITM, DDoS, Phishing, and Malware attacks targeting the processing layer of the IIoT in our hacking scenario-7. In this section, we will explore and examine diverse countermeasures that can be effectively employed to mitigate the impact of these attacks.

MITM Attacks—Countermeasures: To bolster the defense against MITM attacks on the IIoT processing layer, implementing robust access control mechanisms and stringent policies is crucial to thwart unauthorized access to system resources. By carefully regulating user permissions and privileges, malicious actors can be prevented from gaining unauthorized access. Additionally, employing strong encryption methods, such as VPNs, for all Internet access at the network level ensures that sensitive information remains secure and inaccessible to potential eavesdroppers. Alongside encryption, deploying sufficient authentication security measures safeguards against unauthorized users attempting to infiltrate the processing layer. By implementing these countermeasures, the IIoT processing layer can establish a robust line of defense, safeguarding against potential man-in-the-middle attacks and preserving the integrity of critical resources and data [113].

DoS Attacks—Countermeasures: The most effective countermeasures against DDoS on the IIoT processing layer involve the proper implementation and optimization of load balancers. By employing supervised learning classifiers, load balancers can intelligently distribute incoming traffic and identify and mitigate malicious traffic patterns associated with DDoS attacks. Additionally, integrated network monitoring tools provide real-time visibility into network traffic and behavior, enabling prompt detection and response to potential attacks. Through these measures, the processing layer of the IIoT can proactively defend against DDoS attacks, ensuring the uninterrupted and secure operation of critical systems and services [114], [115].

Phishing Attacks—Countermeasures: To bolster defenses against phishing attacks on the IIoT processing layer, several countermeasures can be employed. Digital certificates, U2F authentication, and OTPs can be implemented to enhance the security of user authentication. By utilizing stronger phishing-resistant security keys, the risk of stolen credentials from phishing attempts can be significantly reduced. These measures help to ensure that only authorized users with legitimate credentials can access critical resources, mitigating the potential impact of phishing attacks on the processing layer of the IIoT and safeguarding sensitive information and operations [116], [117], [118].

Malware Attacks—Countermeasures: To counter the threat of malware attacks on the IIoT processing layer and protect against unauthorized access to critical resources, robust security measures must be implemented. Strengthening the security of IIoT devices can be achieved through the adoption of end-point security management solutions tailored for Industry 4.0 companies. By utilizing malware visualization tools in combination with sophisticated classification methods like Nearest Neighbor, Decision Tree, and Random Forest algorithms, potential malware threats can be identified and preemptively mitigated. Additionally, deploying antimalware software enhances the overall defense against malware, ensuring the integrity and security of the IIoT processing layer and safeguarding the smooth operation of essential industrial processes [119], [120], [121].

Countermeasures for Scenario—8: We have discussed the occurrence of Bruteforce attack, Implementation, Statistical, Account hijacking, and Data manipulation attacks targeting the processing layer of the IIoT in our hacking scenario-8. In this section, we will explore and examine diverse countermeasures that can be effectively employed to mitigate the impact of these attacks.

Brute Force Attacks, Implementation Attacks, and Statistical Attacks—Countermeasures: To fortify the IIoT processing layer against brute force attacks, implementation attacks, and statistical attacks, ensuring data protection and confidentiality even after unauthorized access, robust encryption techniques must be implemented for both stored and transmitted data. Utilizing strong encryption methods like AES, RSA, and Triple DES renders stolen data unreadable even if malicious actors gain access. Additionally, employing secure hashing algorithms further enhances data protection. Real-time monitoring is essential to detect any suspicious activities promptly, allowing for swift response and mitigation of potential threats. By combining these countermeasures, the IIoT processing layer can establish a formidable defense against various types of attacks, safeguarding critical data and maintaining a secure and reliable industrial infrastructure [52], [122], [123].

Account Hijacking Attacks—Countermeasures: To bolster the security of the IIoT processing layer and thwart account hijacking attacks that jeopardize identity integrity and lead to personal information theft, robust countermeasures can be employed. Implementing proper data sanitization techniques and data de-identification methods ensure that all PII is removed from the data before transmission and storage on the platform, reducing the risk of exposure. Alternatively, adopting privacy-preserving techniques directly protects sensitive personal information, preventing unauthorized access and misuse. By incorporating these preventive measures, the IIoT processing layer can safeguard user accounts, maintain data privacy, and uphold the integrity of identity information, enhancing overall cybersecurity resilience [124], [125].

Data Manipulation Attacks—Countermeasures: To bolster the defense against data manipulation attacks that seek to compromise data integrity within the IIoT processing layer, several countermeasures can be implemented. Ensuring complete visibility on endpoints and utilizing deep reinforcement learning techniques allow for real-time anomaly detection and rapid

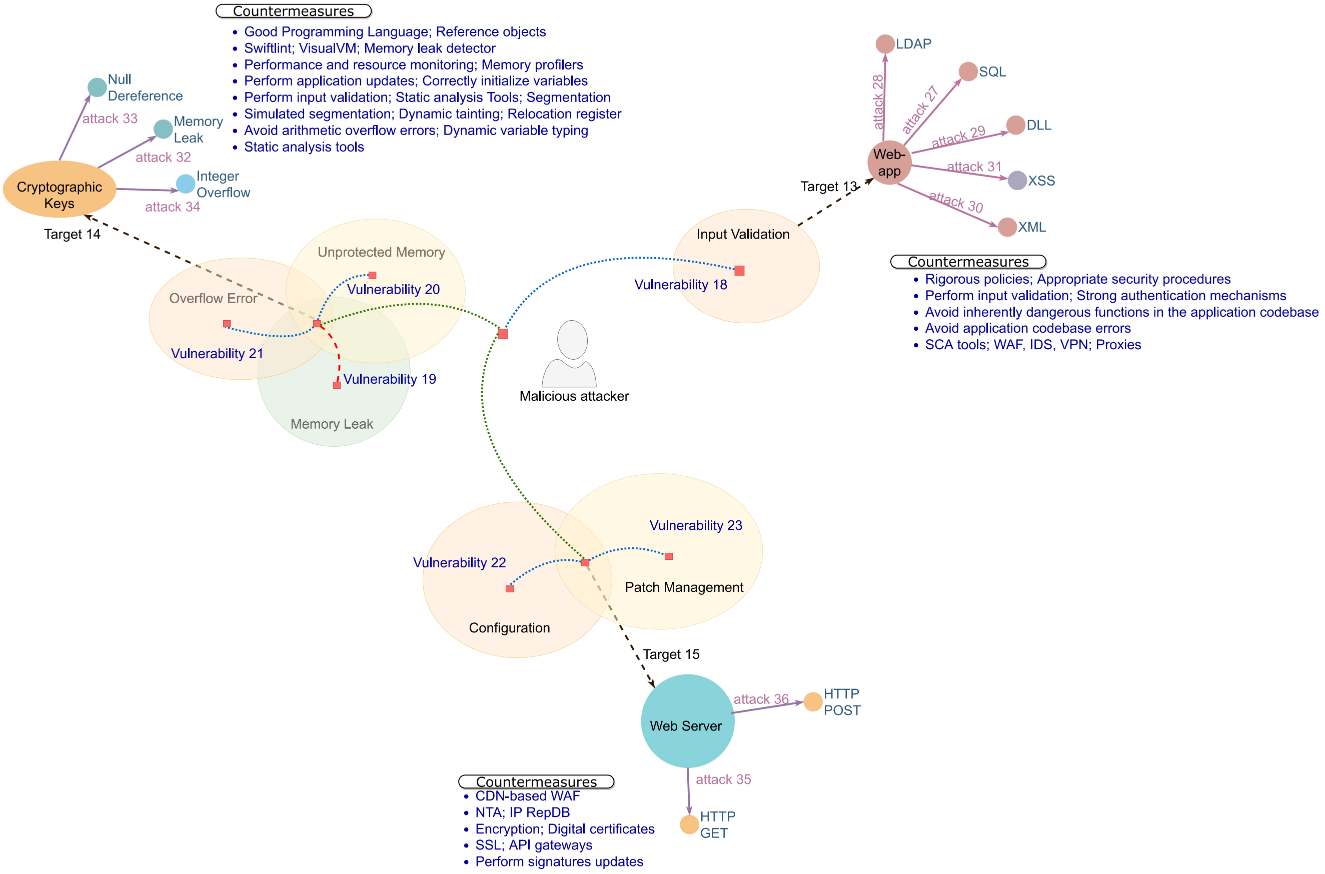

Countermeasures
Good Programming Language; Reference objects
Swiftlint; VisualVM; Memory leak detector
Performance and resource monitoring; Memory profilers
Perform application updates; Correctly initialize variables
Perform input validation; Static analysis Tools; Segmentation
Simulated segmentation; Dynamic tainting; Relocation register
Avoid arithmetic overflow errors; Dynamic variable typing
Static analysis tools
Null Dereference
attack 33
Memory Leak
attack 32
Integer Overflow
attack 34
Cryptographic Keys
Target 14
Unprotected Memory
Overflow Error
Vulnerability 20
Vulnerability 21
Vulnerability 19
Memory Leak
Malicious attacker
Input Validation
Vulnerability 18
Target 13
Web-app
LDAP
SQL
DLL
XSS
XML
attack 28
attack 27
attack 29
attack 31
attack 30
Countermeasures
Rigorous policies; Appropriate security procedures
Perform input validation; Strong authentication mechanisms
Avoid inherently dangerous functions in the application codebase
Avoid application codebase errors
SCA tools; WAF, IDS, VPN; Proxies
Vulnerability 23
Vulnerability 22
Patch Management
Configuration
Target 15
Web Server
attack 36
HTTP POST
attack 35
HTTP GET
Countermeasures
CDN-based WAF
NTA; IP RepDB
Encryption; Digital certificates
SSL; API gateways
Perform signatures updates

Fig. 7. Common vulnerabilities found in this layer, the attacks arising from these vulnerabilities, and the countermeasures to be implemented against these assaults are all described, accordingly, using a hacking scenario on the application layer of IIoT.

response to potential threats. Continuous monitoring of the entire system and network platform ensures proactive identification of suspicious activities. Additionally, data integrity verification tools can be employed to detect and prevent unauthorized data alterations. In instances where data has already been tampered with, file integrity monitoring (FIM) can be employed to restore data integrity and rectify any changes made by malicious actors. These proactive measures collectively fortify the IIoT processing layer, safeguarding data integrity and maintaining the reliability of critical processes within the industrial environment [126], [127], [128].

D. Hacking Scenario on Application Layer

As shown in Fig. 7, this section also uses a hacking scenario on this layer to first highlight the various vulnerabilities that hackers can exploit, then to present the potential attack threats brought on by these vulnerabilities, and finally to discuss the various responses to these cybersecurity incidents.

1) Security Concerns in Prime MotorX's Application Layer: We will explore the potential vulnerabilities in the Prime MotorX network layer that a malicious attacker could exploit to attack a specific target within Prime MotorX.

Scenario−9: The malicious actor may search for injection flaws in Prime MotorX applications using scanners and fuzzers. Upon discovering that some of Prime MotorX's applications allow the transfer of unreliable data to various interpreters, the attacker exploits the lack of proper validation of input and output data to execute a malicious code injection attack. By inserting suspicious code as input into vulnerable applications and modifying the program's execution behavior, the attacker gains unauthorized access, leading to potential malfunctions, disclosure of business information, compromised data integrity, and susceptibility to various cyberattacks, including malware attacks like trojan horses, worms, backdoors, viruses, spyware, and scareware. Specifically, the attacker exploits Web applications vulnerable to SQL Injection attack [129] to gain unauthorized access to the database, compromising sensitive information and system security. Additionally, the attacker launches other code injection attacks, such as LDAP Injection attack [130], [131], DLL Injection attack [132], XML Injection attack [133], [134], [135], and cross-site scripting (XSS) attack [136], [137], [138], to insert malicious code and manipulate user input, further jeopardizing the system's integrity and user credentials. Such code injection attacks pose severe risks to industrial IIoT enterprise applications, potentially leading to data breaches, disclosure of sensitive information, and loss of business profits, which may result in significant financial harm.

Scenario−10: Through forensic analysis of the Prime MotorX applications' memory, the attacker discovers a critical flaw in the ERP application software. The ERP consumes the entire RAM during memory allocation and fails to release the allocated memory even after completing its execution process. Armed with this crucial information, the attacker exploits this vulnerability to launch a Memory-Based Leak attack [139], [140]. By manipulating the memory leak, the attacker can force the ERP application to stop or execute a DoS attack, preventing legitimate users from accessing the application. The vulnerability targeted is the memory leak, focusing on Prime MotorX's ERP application software. Furthermore, the attacker exploits the lack of memory protection on the ERP to gain control over the computer's memory, where the ERP is running. This allows the attacker to corrupt the memory and carry out a Null Pointer Dereference attack [141], another form of memory-based attack that can crash the ERP and facilitate a successful DoS attack. The vulnerability exploited here is the lack of memory protection. Additionally, the attacker can manipulate memory by exploiting program bugs, such as the arithmetic overflow error. By repeatedly attempting multiple overflow attacks, the attacker compromises the ERP application with an Integer Overflow attack [142], [143], another common type of memory-based attack. The vulnerability exploited in this case is arithmetic overflow errors.

Scenario−11: Another potential hacking scenario that a malicious actor may consider involves launching an HTTP flood attack on Prime MotorX's ERP. Assuming the attacker has already gained unauthorized access to multiple devices within Prime MotorX, they exploit a vulnerability related to the improper configuration of content delivery network (CDN) services. The CDN's redirection process, where HTTP requests are sent directly to the Web server without using the CDN cache server, becomes a target for the attacker. By randomly creating multiple new parameters in HTTP GET requests, the attacker bypasses the CDN and overwhelms the Web server with a large number of requests, resulting in a denial of service for legitimate users. Subsequently, the IT administrator attempts to prevent such attacks by implementing signature-based IDS software. However, the attacker finds that, the IDS software is outdated due to improper security patch updates. Taking advantage of this weakness, the attacker evades the IDS and launches a hypertext transfer protocol POST attack on the Web server, overloading it with continuous POST requests and data from HTML forms. The exploited vulnerability is poor patch management. HTTP flood attacks, both HTTP GET and HTTP POST [144], [145], [146], [147], pose serious threats to IIoT businesses, leading to financial losses, reputational damage, revenue loss, and decreased consumer trust.

2) Countermeasures for Security Issues in the Application Layer: According to the examination of the various weaknesses and assaults concerning the application layer as formerly detailed in scenarios 9–11, we discuss in this section the relevant countermeasures that must be put in place to secure the application layer of the IIoT.

Countermeasures for Scenario−9: We have discussed the occurrence of lack of input validation, SQL Injection, LDAP Injection, DLL Injection, XML Injection, and XSS attacks which are all classified as malicious code injection attacks targeting the application layer of the IIoT in our hacking scenario9. In this section, we will explore and examine diverse countermeasures that can be effectively employed to mitigate the impact of these attacks.

Malicious Code Injection Attacks−Countermeasures: To mitigate the risk of various malicious code injection attacks, such as SQL injection, LDAP injection, DLL injection, XML injection, and XSS attacks, which target the IIoT application layer, robust countermeasures must be implemented. Institutions can establish comprehensive security policies and procedures during application development to enforce strict input validation practices. This includes employing pattern matching and cryptographic measures to ensure the integrity of data inputs. Utilizing AI and ML techniques can aid in identifying and thwarting injection attacks, while continuous monitoring helps detect and change attack payload patterns. Access to application source code should be restricted to authorized personnel only, and inherently dangerous functions in the codebase should be avoided. Employing static code analysis (SCA) tools allows for the identification and removal of potential malicious code from the application source code. Enhancing application and API security through Web application firewalls (WAFs), IDSs, VPNs, and proxies strengthens the overall protection against these attacks. By implementing these comprehensive measures, IIoT applications can be safeguarded against the threat of code injection attacks, preserving the integrity and security of critical industrial systems [148], [149], [150], [151], [152], [153], [154].

Countermeasures for Scenario−10: We have discussed the occurrence of Memory-Based Leak, Null Pointer Dereference, and Integer Overflow attacks targeting the application layer of the IIoT in our hacking scenario 10. In this section, we will explore and examine diverse countermeasures that can be effectively employed to mitigate the impact of these attacks

Memory-Based Leak Attacks−Countermeasures: Defending against memory-based leak attacks on the IIoT application layer requires a multifaceted approach. First and foremost, developers must focus on writing robust and bug-free code during the application development process. This involves a comprehensive understanding of memory management features and the use of reference objects. Utilizing tools like Swift Lint, VisualVM, or memory leak detectors enables developers to identify and rectify early memory leaks and flaws in the application code. Additionally, continuous monitoring of running applications using performance and resource monitors, such as memory profilers, aids in the timely detection and elimination of potential memory leaks. Keeping applications updated with the latest patches and security fixes further strengthens their resilience against memory-based leak attacks. By implementing these countermeasures, the IIoT application layer can be better safeguarded against memory-based leak attacks, ensuring the integrity and reliability of critical industrial systems [155], [156], [157], [158], [159].

Null Pointer Dereference Attacks−Countermeasures: To defend against NULL pointer dereference attacks on the IIoT application layer, certain countermeasures are essential during development and implementation. Developers should avoid using NULL pointers in the code, opting for programming

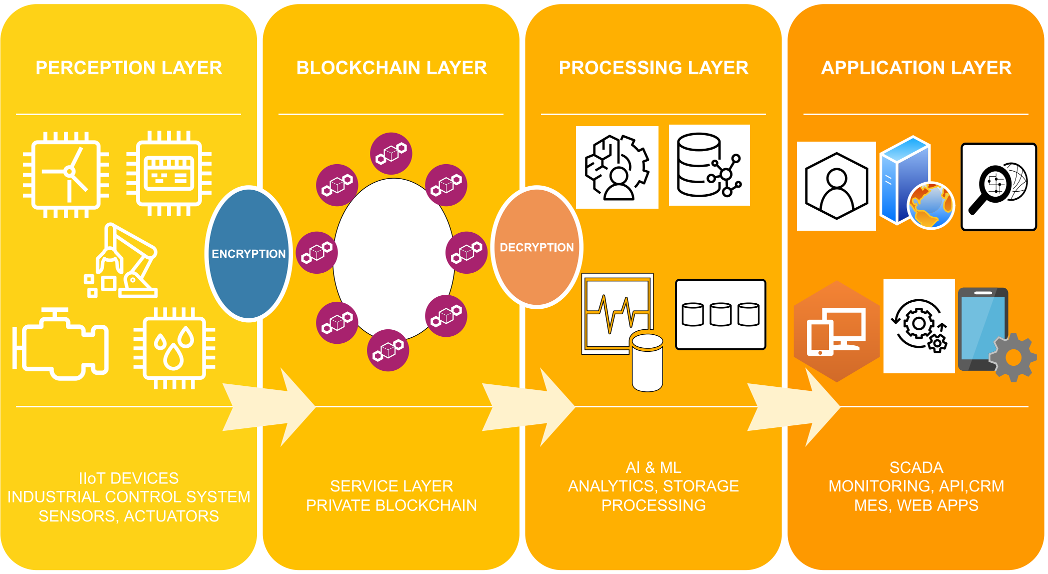


Fig. 8. Example of a typical IIoT blockchain architecture.

languages that minimize NULL pointer dereference flaws. Proper initialization of variables and data objects should be ensured to nonnull values. Managing reference objects carefully to avoid pointing to memory address 0 is crucial. Rigorous input validation can prevent malicious input from exploiting NULL pointer vulnerabilities. Utilizing static analysis tools helps detect and address null pointers before and after memory deallocation, reducing risks. Strengthening memory security through access control measures, such as segmentation, guard key, and capability-based addressing enhances memory protection against NULL pointer dereference attacks, ensuring the reliability of IIoT applications [160].

Integer Overflow Attacks—Countermeasures: Defending against integer overflow attacks at the IIoT application layer involves proactive measures in the code development process and implementing detection mechanisms. Developers should meticulously craft their code to avoid arithmetic overflow errors, leveraging appropriate programming language features with dynamic variable typing to prevent or mitigate integer overflow vulnerabilities. By ensuring thorough input validation on all integer inputs, potential risks of integer overflow can be minimized. Additionally, incorporating static analysis tools within the application development process aids in detecting and addressing integer overflow vulnerabilities promptly. By employing these countermeasures, the IIoT application layer can enhance its resilience against integer overflow attacks, safeguarding the integrity and stability of the system [61], [155], [157], [160].

Countermeasures for Scenario—11: We have discussed the occurrence of HTTP GET, and HTTP POST attacks which are all classified as HTTP flooding attacks, targeting the application layer of the IIoT in our hacking scenario 11. In this section, we will explore and examine diverse countermeasures that can be effectively employed to mitigate the impact of these attacks.

HTTP Flooding Attacks—Countermeasures: To defend against HTTP flooding attacks, which include HTTP GET and POST attacks that aim to overwhelm services and hinder legitimate user access, comprehensive preventive and protective security measures must be implemented throughout the IIoT application layer. Combining a CDN with a WAF can effectively detect and block suspicious HTTP requests, ensuring server-side applications are inaccessible to malicious requests. Utilizing network traffic analysis and an IP reputation database (IP RepDB) enables continuous monitoring of malicious Web traffic, facilitating timely detection and analysis. Additionally, employing encryption, such as SSL certificates, for sensitive PII sent via HTML forms adds an extra layer of protection. Regular WAF updates, driven by automatic software updates from vendors, enhance defense against HTTP flood attacks. Adopting API gateways like Auth0 enforces security checks, requiring user validation and authentication before accessing applications, further mitigating the risk of HTTP flooding attacks. These countermeasures collectively bolster the IIoT application layer's resilience against HTTP flooding attacks, ensuring robust and secure service availability for legitimate users [161], [162], [163], [164], [165].

IV. Privacy for IIoT Systems Based on Blockchain

We previously covered various defenses to prevent, detect, and eliminate potential threats that might jeopardize security at every level of the IIoT architecture. The countermeasures addressed in Section III are further extended in this section to other existing IIoT privacy solutions, including but not limited to the blockchain.

Blockchain can overcome privacy concerns when combined with encryption, consensus methods, and distributed data storage in the IIoT [166]. Fig. 8 shows a representation of a typical blockchain architecture for the IIoT. Blockchain enables identity protection in IIoT environments like industrial Edge computing or cloud computing [167]. Public keys (PKIs) within the blockchain grant access to users, and encrypted transactions using the receiver's PKI protect sensitive information. ECC ensures data security in the IIoT ecosystem [168].

Wang et al. [169] proposed a blockchain-based privacy-sharing scheme in IIoT to protect keys, ensure data consistency, and incentivize communication. Chen et al. [170] used Hyperledger Fabric to protect sensitive data during enterprise data sharing. Feng et al. [171] utilized zero-knowledge proof to maintain data privacy in IIoT data sharing. Wang et al. [172] applied identity-based signatures and homomorphic encryption for privacy protection in energy trading. Bao et al. [173] developed an identity management strategy to preserve user privacy. Rahulamathavan et al. [174] proposed an end-to-end privacy-preserving IoT blockchain architecture. Jiang et al. [175] enhanced search performance with privacy using Bloom filters. Feng et al. [176] utilized blockchain and homomorphic encryption for secure IIoT data communication.

Zhao et al. [177] explored the integration of blockchain and IIoT, addressing applications and challenges. Zhao et al. [178] proposed a dynamic reputation management system in blockchain-based crowdsensing. Ernest and Shiguang [168] addressed security challenges using ECC in blockchain for IIoT with edge computing.

V. Discussion and Analogy With Recent Surveys

More businesses are implementing IIoT technology to increase automation and workflow efficiency and lower operational risk to maximize industrial productivity. However, many industrial organizations have encountered security issues in the IIoT, which has exposed them to various cyberattacks. Thus, extensive studies have been conducted on how to secure the IIoT against any potential attacks.

This article draws inspiration from existing studies on security and privacy protection in IIoT systems, unlike other surveys that simply provide a comprehensive overview of the many security threats, weaknesses, and responses for each level of the IIoT architecture. Our research study, on the other hand, establishes a set of concrete hacking scenarios on different layers of the IIoT to accurately describe and comprehend the entire process of how an adversary gathers information about system flaws and initiates attack vectors to achieve his goals, which are theft, vandalism, and sabotage. The first phase in our layer-based hacking scenario is to identify potential vulnerabilities. Furthermore, our hacking scenarios provide a variety of targets from which the hacker can launch his attacks. Once the target has been determined, the opponent can commence its attack. The suggested hacking scenario examines all of the outcomes, immediate effects, and associated countermeasures that follow such security attacks. As shown in Figs. 4–7, our layer-based hacking scenarios list the security vulnerabilities, estimated targets, potential threats, consequences of these attacks, and possible solutions in an IIoT architecture. The entire summary is provided in Table II. Given the necessity of protecting users' privacy when their credentials and PII are transferred to the application layer or managed by a third party such as service providers, our research includes existing privacy preservation methods based on the blockchain in IIoT to address this issue.

VI. Open Issues and Future Research Perspectives

As mentioned earlier, we reviewed several existing security and privacy solutions in the IIoT with a particular focus on blockchain-based IIoT systems. However, these methods present some limitations, as most of them are still in the experimental stage and potentially open to security and privacy issues. The main problems with traditional blockchain and privacy techniques in the IIoT are outlined in this section. Then, to address these challenges, the report develops new approaches and strategies that can be employed in future research to strengthen security and enhance privacy protection in the IIoT to address these challenges.

A. Formulation of Problems

When data is transferred directly to the cloud, technology such as blockchain can effectively solve privacy issues of edge and cloud computing-enabled IIoT in a sense where businesses can still use third-party cloud services without compromising the privacy of their data.

1) *Problem−1 (Pattern Recognition Among PKIs):* Traditional Blockchain based on PKI random generation (PKRG), which uses secret keys and random nonces to generate randomly new PKIs, can indeed ensure user anonymity and untraceability in IIoT systems [168]. However, a malicious actor who already belongs to the blockchain may be able to examine the history of previous PKIs to distinguish patterns and find some known knowledge if the random distribution among keys is closed, then make connections between addresses and eventually disclose the identity of users. Suppose P_1 and P_2 are both participant members in a blockchain BC and let div_{P_1} and div_{P_2} be the respective IIoT devices of P_1 and P_2. Each $(div_{P_1}, div_{P_2}) \in BC$ is identified by its PKI PK_{P_1} and PK_{P_2}, where every transaction information is public. By examining the data, an interested group G might be able to distinguish patterns and make connections between addresses and eventually disclose the identities of P_1 and P_2.
2) *Problem−2 (Insecure Against User and Server Impersonation Attacks):* The second problem is that the previous blockchain methods earlier mentioned in Section IV do not take into account the semantics of low-frequency keywords and can be insecure against known session-specific temporary information attacks, 51% attacks, and user and server impersonation attacks [179], [180], [181].

Differential privacy, which is a privacy-preserving technique, can be integrated with the blockchain to ensure the privacy of individuals in cases where data gathered and processed in the cloud can be published from the cloud to the public for commercial purposes.

1) *Problem−3 (Optimization):* Existing differential privacy methods, such as methods [182], [183], can set a privacy probability threshold to select the best privacy parameter based on the Laplace distribution and Privacy Fault Tolerance and, thus, achieve privacy protection while ensuring data utility. However, the algorithm often

TABLE II
SUMMARY OF THE HACKING SCENARIO ON ATTACKS, VULNERABILITIES, EFFECTS, AND COUNTERMEASURES ON THE IIOT ARCHITECTURE SYSTEM

IIoT Layers	*Attacks*	*Vulnerabilities*	*Effects*	*Countermeasures*
Perception Layer	• Physical damage • Tampering • Ransomware • Malware • Default password • Brute-force • DoS	• Poor hardening • Patch management failure • Weak authentication • Misconfiguration • Legacy devices	• Device failure • Material damage • Shut down machinery processes • Causing life-threatening injuries • Data theft and data loss • Trigger other cyberattacks	**Physical Security Measures:** • Use strong fences, anti-theft, and tracking devices. • Install motion detection cameras and alarm systems • Conduct regular inspections and assessment of equipment **Device and Software Security:** • Implement anti-tampering and anti-counterfeiting measures. • Update & patch regularly • Use cyber threat hunting (CTH) and secure endpoint **Device and Network Security:** • Employ firewall, IDS, MFA, IPFS, and OTP • Perform regular data backup **Compliance:** • Follow laws and regulations on device management .
Network Layer	• Eavesdropping • Traffic analysis • Replay • Wormhole • Blackhole • Sybil • Sinkhole • Routing • DoS/DDoS	• Insufficient data encryption • Insufficient transport encryption • Misconfigured authentication • Outdated software • Firewall issues • Weak encryption	• Disrupt and slow down production • Cause bandwidth bottlenecks • Data breach, data theft • privacy breach • Cause unauthorized access • Compromise data integrity • Financial loss	**Authentication:** • Deploy authentication, firewalls, WPA3, VPN, and encryption **Network Security:** • Use NAT, TLS, SSL, tokens, traffic monitoring, and session key **Data Validation:** • Apply timestamps, AODV, IDS, IPS, RPL, PDR, and RSS location validation **Packet Handling:** • Use geographical packet leashes method and send protocol **Modulation:** • Use modulation packet, set, ML, TECP intercept, ingress, and egress filtering **Threat Protection:** • Use anti-spoofing, anti-virus, and anti-malware **Performance and Maintenance:** • Perform checks, increase bandwidth, and keep systems patched
Processing Layer	• MITM • DDoS • Phishing • Malware • Brute Force • Ransomware • Default Password • Phishing • Statistical Attacks • Account Hijacking • Data Manipulation	• Lack of Web Traffic Encryption • Misconfigured Load Balancing • Poor Identity Management • Poor Endpoint Management • Poor Data Encryption • Lack of Data Sanitization	• Data Breach • Data Theft • Steal User Credentials • Unauthorized Acquisitions • Slow Down and Disrupt Production • Cause Services to Malfunction • Compromise Data Integrity • Cause Financial Loss • Defamation	**Data Protection Measures:** • Use encryption, VPN, authentication • Use AES, RSA, Triple DES, secure hashing algorithms **Network Security:** • Optimize load balancers, real-time monitoring, digital certificates **User Authentication and Access:** • Use U2FA, OTP, phishing-resistance security keys, endpoint security **Malware Defense:** • Employ malware visualization tools with classification methods **Endpoint Security and Data Privacy:** • Visibility on endpoints, use FIM, data sanitization, data de-identification **Data Integrity and AI Security Measures:** • Deploy data integrity verification tools, deep reinforcement learning • Use supervised learning classifiers
Application Layer	• SQL Injection • LDAP Injection • DLL Injection • XML Injection • Null Dereference • Integer Overflow • HTTP Flood (GET and POST)	• Lack of Input/Output Data Validation • Memory Leak • Lack of Memory Protection • Arithmetic Overflow Errors • Improper Configuration • Poor Patch Management	• Applications and Software Failure • Triggered Cyberattacks • Compromised Data Integrity • Data Breach • Privacy Breach • Financial Loss • Unauthorized Acquisitions • Reputational Damage • Loss of Consumer Trust	**Policies & Regulations on Application:** • Perform Input Validation • Use Pattern Matching • Encryption • Choose an Appropriate Programming Language **Web Application Firewall (WAF) and Network Security:** • Use WAF, IDS, VPN • Use WAF, CDN-Based WAF, IDS, VPN • Use Proxies, IP RepDB , API Gateways, API Gateways **Memory and Code Protection:** • Deploy Guard Keys & Rings , Paged Virtual Memory, Dynamic Tainting • Deploy Relocation-Based Addressing, SCA (Static Code Analysis) • Simulated Segmentation, Memory Protection Techniques and Memory Filters **Memory Leak Detection:** • Use Reference Object , Swift Lint, VisualVM • Memory Detectors , Python Memory Leak Detectors • Python Memory Leak Detectors **Memory Leak Detection:** • Avoid Inherent Function in Application Code • Avoid Inherent Function in Application Code

suffers from optimization problems in finding the best privacy parameter ϵ.

2) *Problem−4 (The Integration of IIoT Systems):* Most organizations struggle to integrate IIoT security and privacy solutions into their overall system since there is no equipped and adapted design model to utilize as a unique framework to provide security and privacy for both industrial edge and cloud computing.

It is debatable whether conventional cyber defenses can keep up with contemporary cyber threats. Unfortunately, for two reasons in particular variety and space, this is not the case.

1) *Problem−5 (The Variety):* Due to the variety and rapid development of new IIoT technologies, cloud environments, API, evolving digital transformation mechanisms, loads of digital infrastructure to monitor, etc., businesses are finding it more and more challenging to have complete visibility over their operations and protect them from cyber-attacks.

2) *Problem−6 (The Pace):* Due to the rapid pace at which digital estates are expanding, as well as the rapidity and dynamism with which industrial machines, devices, and end users are networked and communicating, as a result, the IoT infrastructure develops vulnerabilities

that frequently result in brand-new ways to compromise security that conventional security solutions are unable to prevent.

B. Proposed Solutions

This part covers the various approaches and strategies that have been proposed to overcome the open security and privacy challenges outlined above.

1) A robust IIoT security framework relies on minimizing pattern recognition among PKIs and strengthening protection against session-specific attacks, user impersonation, and server impersonation. To safeguard user identities, organizations should employ secure encryption by generating stronger PKIs. For instance, regulating the random distribution of PKIs within blockchain transactions using the elliptic curve digital signature algorithm (ECDSA) ensures the creation of new keys considering previous distributions. By designing a blockchain security approach controlling the randomness between previous and newly generated PKIs, IIoT devices can utilize updated keys for each transaction, enhancing pattern complexity and deterring intruders from easily guessing keys, thus improving the security of existing blockchain-based IIoT.
2) We can even combine the suggested ECDSA, which the blockchain utilizes to generate new PKIs, with the enhanced ECC-based three-factor multiserver authentication approach proposed by Wu et al. [184] to give an even stronger and better security mechanism for IIoT systems.
3) To address the problem of optimization in determining the best privacy parameter ϵ for Blockchain technology that incorporates differential privacy mechanisms, we suggest that researchers develop new methods for blockchain to achieve security based on optimization algorithms and differential privacy when industrial data is publicly published. We can, for example, utilize an optimization approach like stochastic gradient descent (SGD) [185], which may aid in determining the ideal privacy parameter ϵ by executing parameter changes for each training sample while ensuring data usefulness. Additionally, businesses might consider developing and designing solutions that will assist in the integration and adaptation of this new type of security and privacy mechanisms in IIoT industries.
4) We also urge the scientific community to play a bigger role in the creation and application of a more adaptable unified IoT security architecture, which will serve as the central point of security and privacy for all IIoT layers and enable any industry 4.0 company using IIoT technology to predict, identify, and eliminate cybersecurity issues or threats from the edge to end users in real-time.
5) Researchers and organizations need to come up with innovative approaches to protect IIoT companies against modern cybersecurity threats brought on by the pace and variety of IIoT. We can leverage AI and develop solutions integrated with blockchain to not only provide proportionate responses and smart decisions to act against new threats as they arise but also to be able to instantly and globally guarantee the security of the entire business.

VII. Conclusion

Industry 4.0 technologies, including IIoT, enable smart production by empowering enterprises to prioritize long-term business goals above minimal risks. The IIoT represents a significant paradigm shift in how various businesses operate. It employs integrated sensors to keep track of every stage of business activity as it occurs. This expedites dataflow and analysis, resulting in quick insight, which improves decision making, corrective action, and enables predictive maintenance. With all of this in place, industrial companies such as smart factories may increase production, improve quality, and cut costs. However, as IIoT-enabling technologies and applications become more prevalent, security concerns and privacy protection issues become more difficult to manage, leading to cyber-physical risks. The usage of legacy equipment, the complexity of interconnected heterogeneous smart devices, and improper and insecure IoT implementations in ICS are all security challenges that render IIoT more vulnerable to cyber threats.

The primary factors that inspired us to write this review are to understand and identify how and when these security and privacy vulnerabilities emerge, as well as what responses to take against them. In this article, we give a comprehensive overview of the literature on security and privacy concerns, vulnerabilities, threats, and corresponding responses in the IIoT. To help the reader understand, we first introduce what the reader needs to know about IIoT in Industry 4.0. We created a taxonomy that categorizes the complete layered modular architecture of IIoT. Then, this study looks closely at a variety of security issues involving IIoT systems. On different tiers of the IIoT, a hacking scenario is employed as a methodology to accurately explain and comprehend the full process of how an adversary acquires information about system defects and launches attack vectors. Identifying potential vulnerabilities is the first step in our layer-based hacking scenario. Additionally, our hacking scenarios offer a choice of targets from which the hacker can launch his attacks. The hypothetical hacking scenario looks at all of the outcomes, immediate repercussions, and related countermeasures that follow such security attacks. The next step is to provide a full analysis of various IIoT privacy solutions built on blockchain. This article also outlines the several significant issues that traditional blockchains and privacy-protection strategies face. Finally, we propose novel ideas and methodologies that may enhance security and privacy protection in the IIoT system in the hopes of inspiring other scholars to make significant contributions to the advancement of securing the IIoT.

Author Contributions

Anselme Herman Eyeleko methodology, Tao Feng validation, Anselme Herman Eyeleko investigation and formal analysis, Anselme Herman Eyeleko original draft preparation, Anselme Herman Eyeleko and Tao Feng writing, review and editing, Anselme Herman Eyeleko visualization, Tao Feng supervision, Tao Feng resources and funding acquisition.

References

[1] E. Weber, M. Büttgen, and S. Bartsch, "How to take employees on the digital transformation journey: An experimental study on complementary leadership behaviors in managing organizational change," *J. Bus. Res.*, vol. 143, pp. 225–238, Apr. 2022.

[2] J. Kaur, N. Santhoshkumar, M. Nomani, D. K. Sharma, J. P. Maroor, and V. Dhiman, "Impact of Internet of Things (IoT) in retail sector," *Mater. Today Proc.*, vol. 51, pp. 26–30, Jan. 2022.

[3] J. Wang, M. K. Lim, C. Wang, and M.-L. Tseng, "The evolution of the Internet of Things (IoT) over the past 20 years," *Comput. Ind. Eng.*, vol. 155, May 2021, Art. no. 107174.

[4] V. R. Kebande, "Industrial Internet of Things (IIoT) forensics: The forgotten concept in the race towards industry 4.0," *Forensic Sci. Int. Rep.*, vol. 5, Jul. 2022, Art. no. 100257.

[5] F. Li et al., "A measurement study on device-to-device communication technologies for IIoT," *Comput. Netw.*, vol. 192, Jun. 2021, Art. no. 108072.

[6] C. W. Chen, "Internet of Video Things: Next-generation IoT with visual sensors," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 6676–6685, Aug. 2020.

[7] L. Zhao, I. B. M. Matsuo, Y. Zhou, and W.-J. Lee, "Design of an Industrial IoT-based monitoring system for power substations," *IEEE Trans. Ind. Appl.*, vol. 55, no. 6, pp. 5666–5674, Nov./Dec. 2019.

[8] P. Moens et al., "Scalable fleet monitoring and visualization for smart machine maintenance and Industrial IoT applications," *Sensors*, vol. 20, no. 15, p. 4308, 2020.

[9] J. Rosales, S. Deshpande, and S. Anand, "IIoT based augmented reality for factory data collection and visualization," *Procedia Manuf.*, vol. 53, pp. 618–627, Jan. 2021.

[10] A. Corallo, M. Lazoi, M. Lezzi, and A. Luperto, "Cybersecurity awareness in the context of the Industrial Internet of Things: A systematic literature review," *Comput. Ind.*, vol. 137, May 2022, Art. no. 103614.

[11] M. Abdel-Basset, H. Hawash, and K. Sallam, "Federated threat-hunting approach for microservice-based industrial cyber-physical system," *IEEE Trans. Ind. Informat.*, vol. 18, no. 3, pp. 1905–1917, Mar. 2022.

[12] P. Jayalaxmi, R. Saha, G. Kumar, N. Kumar, and T.-H. Kim, "A taxonomy of security issues in Industrial Internet-of-Things: Scoping review for existing solutions, future implications, and research challenges," *IEEE Access*, vol. 9, pp. 25344–25359, 2021.

[13] P. Malhotra, Y. Singh, P. Anand, D. K. Bangotra, P. K. Singh, and W.-C. Hong, "Internet of Things: Evolution, concerns and security challenges," *Sensors*, vol. 21, no. 5, p. 1809, 2021.

[14] E. Lara, L. Aguilar, M. A. Sanchez, and J. A. García, "Lightweight authentication protocol for M2M communications of resource-constrained devices in Industrial Internet of Things," *Sensors*, vol. 20, no. 2, p. 501, 2020.

[15] I. Mugarza, J. L. Flores, and J. L. Montero, "Security issues and software updates management in the Industrial Internet of Things (IIoT) era," *Sensors*, vol. 20, no. 24, p. 7160, 2020.

[16] G. McDonald, P. Papadopoulos, N. Pitropakis, J. Ahmad, and W. J. Buchanan, "RansomWare: Analysing the impact on windows active directory domain services," *Sensors*, vol. 22, no. 3, p. 953, 2022.

[17] U. Hunkeler, H. L. Truong, and A. Stanford-Clark, "MQTT-S—A publish/subscribe protocol for wireless sensor networks," in *Proc. IEEE 3rd Int. Conf. Commun. Syst. Softw. Middleware Workshops (COMSWARE)*, 2008, pp. 791–798.

[18] A. Karmakar, N. Dey, T. Baral, M. Chowdhury, and M. Rehan, "Industrial Internet of Things: A review," in *Proc. IEEE Int. Conf. Opto-Electron. Appl. Opt. (Optronix)*, 2019, pp. 1–6.

[19] H. ElMaraghy, L. Monostori, G. Schuh, and W. ElMaraghy, "Evolution and future of manufacturing systems," *CIRP Ann.*, vol. 70, no. 2, pp. 635–658, 2021.

[20] K. O. M. Salih, T. A. Rashid, D. Radovanovic, and N. Bacanin, "A comprehensive survey on the Internet of Things with the industrial marketplace," *Sensors*, vol. 22, no. 3, p. 730, 2022.

[21] M. Alom and S. E. Kesen, "Smart warehouses in logistics 4.0," in *Logistics 4.0*. Boca Raton, FL, USA: CRC Press, 2020, pp. 186–201.

[22] L. Rizkallah, N. Potter, K. Reed, D. Reynolds, M. Salman, and S. Bhunia, "Red toad, blue toad, hacked toad?" in *Proc. IEEE World AI IoT Congr. (AIIoT)*, 2022, pp. 379–386.

[23] A. Dwyer, "The NHS cyber-attack: A look at the complex environmental conditions of WannaCry," *RAD Mag.*, vol. 44, no. 512, pp. 25–26, 2018.

[24] T. Roccia, *Triton Malware Spearheads Latest Attacks on Industrial Systems*, McAfee, San Jose, CA, USA, 2018.

[25] S. Abaimov and M. Martellini, *Cyber Arms: Security in Cyberspace*. Boca Raton, FL, USA: CRC Press, 2020.

[26] P. K. Malik et al., "Industrial Internet of Things and its applications in industry 4.0: State of the art," *Comput. Commun.*, vol. 166, pp. 125–139, Jan. 2021.

[27] G. Fiore, "From manufacturing-as-a-service to cloud manufacturing: Real world cases analysis in tube and pipe fabrication industry," M.S. thesis, School Ind. Eng., POLIMI Univ., Milan, Italy, 2019.

[28] G. Lechowski and M. Krzywdzinski, "Emerging positions of German firms in the Industrial Internet of Things: A global technological ecosystem perspective," *Global Netw.*, vol. 22, no. 4, pp. 666–683, 2022.

[29] K. Sekar, S. A. Shah, and A. A. Athithan, "Reviewing the challenges in maintaining the reliability and accuracy of IoT systems for remaining useful life prediction," *IOP Conf. Mater. Sci. Eng.*, vol. 1128, no. 1, 2021, Art. no. 12010.

[30] S. Bagchi et al., "New Frontiers in IoT: Networking, systems, reliability, and security challenges," *IEEE Internet Things J.*, vol. 7, no. 12, pp. 11330–11346, Jul. 2020.

[31] A. Rayes, S. Salam, A. Rayes, and S. Salam, "Internet of Things (IoT) overview," in *Internet of Things From Hype to Reality: Road to Digitization*. Cham, Switzerland: Springer, 2017, pp. 1–34.

[32] L. Gant. "Airworthiness directives: The Boeing company airplanes." 2022. [Online]. Available: https://public-inspection.federalregister.gov/2022-03967.pdf

[33] A. Rafiq, P. Wang, M. Wei, M. S. A. Muthanna, and N. N. Josbert, "Mitigation impact of energy and time delay for computation offloading in an Industrial IoT environment using Levenshtein distance algorithm," *Security Commun. Netw.*, vol. 2022, Feb. 2022, Art. no. 6469380.

[34] K. Kaur, S. Garg, G. S. Aujla, N. Kumar, J. J. Rodrigues, and M. Guizani, "Edge computing in the Industrial Internet of Things environment: Software-defined-networks-based edge-cloud interplay," *IEEE Commun. Mag.*, vol. 56, no. 2, pp. 44–51, Feb. 2018.

[35] X. Dai et al., "Task co-offloading for D2D-assisted mobile edge computing in Industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 19, no. 1, pp. 480–490, Jan. 2023.

[36] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial Internet of Things: Challenges, opportunities, and directions," *IEEE Trans. Ind. Informat.*, vol. 14, no. 11, pp. 4724–4734, Nov. 2018.

[37] A. A. Mirani, G. Velasco-Hernandez, A. Awasthi, and J. Walsh, "Key challenges and emerging technologies in Industrial IoT architectures: A review," *Sensors*, vol. 22, no. 15, p. 5836, 2022.

[38] S. J. Moore, C. D. Nugent, S. Zhang, and I. Cleland, "IoT reliability: A review leading to 5 key research directions," *CCF Trans. Pervasive Comput. Interact.*, vol. 2, pp. 147–163, Aug. 2020.

[39] S. N. Deshpande and R. M. Jogdand, "A survey on Internet of Things (IoT), Industrial IoT (IIoT) and industry 4.0," *Int. J. Comput. Appl.*, vol. 175, no. 27, pp. 20–27, 2020.

[40] P. Manna and R. K. Das, "Scalability in Internet of Things: Techniques, challenges and solutions," *Int. J. Res. Eng. Appl. Manag.*, vol. 7, no. 1, pp. 259–261, 2021.

[41] Y. Wu, H.-N. Dai, and H. Wang, "Convergence of blockchain and edge computing for secure and scalable IIoT critical infrastructures in industry 4.0," *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2300–2317, Feb. 2021.

[42] S. V. Jardim, "The electronic health record and its contribution to healthcare information systems interoperability," *Procedia Technol.*, vol. 9, pp. 940–948, Jan. 2013.

[43] H. Kayan, M. Nunes, O. Rana, P. Burnap, and C. Perera, "Cybersecurity of industrial cyber-physical systems: A review," *ACM Comput. Surveys*, vol. 54, no. 11s, pp. 1–35, 2022.

[44] U. Urooj, B. A. S. Al-Rimy, A. Zainal, F. A. Ghaleb, and M. A. Rassam, "Ransomware detection using the dynamic analysis and machine learning: A survey and research directions," *Appl. Sci.*, vol. 12, no. 1, p. 172, 2021.

[45] H. Oz, A. Aris, A. Levi, and A. S. Uluagac, "A survey on ransomware: Evolution, taxonomy, and defense solutions," *ACM Comput. Surveys*, vol. 54, no. 11s, pp. 1–37, 2022.

[46] A. K. Sood, S. Zeadally, and R. Bansal, "Exploiting trust: Stealthy attacks through socioware and insider threats," *IEEE Syst. J.*, vol. 11, no. 2, pp. 415–426, Jun. 2017.

[47] C. Xenofontos, I. Zografopoulos, C. Konstantinou, A. Jolfaei, M. K. Khan, and K.-K. R. Choo, "Consumer, commercial, and Industrial IoT (in) security: Attack taxonomy and case studies," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 199–221, Jan. 2022.

[48] K. Nimmy, S. Sankaran, K. Achuthan, and P. Calyam, "Securing remote user authentication in Industrial Internet of Things," in *Proc. IEEE 19th Annu. Consum. Commun. Netw. Conf. (CCNC)*, 2022, pp. 244–247.

[49] P. Chen, S. Liu, B. Chen, and L. Yu, "Multi-agent reinforcement learning for decentralized resilient secondary control of energy storage systems against DoS attacks," *IEEE Trans. Smart Grid*, vol. 13, no. 3, pp. 1739–1750, May 2022.

[50] A. Mihoub, O. B. Fredj, O. Cheikhrouhou, A. Derhab, and M. Krichen, "Denial of service attack detection and mitigation for Internet of Things using looking-back-enabled machine learning techniques," *Comput. Elect. Eng.*, vol. 98, Mar. 2022, Art. no. 107716.

[51] L. Bošnjak, J. Sreš, and B. Brumen, "Brute-force and dictionary attack on hashed real-world passwords," in *Proc. 41st Int. Conv. Inf. Commun. Technol. Electron. Microelectron. (MIPRO)*, 2018, pp. 1161–1166.

[52] R. Verma, N. Dhanda, and V. Nagar, "Enhancing security with in-depth analysis of brute-force attack on secure hashing algorithms," in *Proc. Trends Electron. Health Informat. (TEHI)*, 2022, pp. 513–522.

[53] J. Huck and F. Breitinger, "Wake up digital forensics' community and help combat ransomware," *IEEE Security Privacy*, vol. 20, no. 4, pp. 61–70, Jan. 2022.

[54] T. Reshmi, "Information security breaches due to ransomware attacks—A systematic literature review," *Int. J. Inf. Manag. Data Insights*, vol. 1, no. 2, 2021, Art. no. 100013.

[55] X. Yang, L. Shu, Y. Liu, G. P. Hancke, M. A. Ferrag, and K. Huang, "Physical security and safety of IoT equipment: A survey of recent advances and opportunities," *IEEE Trans. Ind. Informat.*, vol. 18, no. 7, pp. 4319–4330, Jul. 2022.

[56] M.-C. Lu, Q.-X. Huang, M.-Y. Chiu, Y.-C. Tsai, and H.-M. Sun, "PSPS: A step toward tamper resistance against physical computer intrusion," *Sensors*, vol. 22, no. 5, p. 1882, 2022.

[57] S. Alyahya, W. U. Khan, S. Ahmed, S. N. K. Marwat, and S. Habib, "Cyber secure framework for smart agriculture: Robust and tamper-resistant authentication scheme for IoT devices," *Electronics*, vol. 11, no. 6, p. 963, 2022.

[58] A. Maurushat and K. Nguyen, "The legal obligation to provide timely security patching and automatic updates," *Int. Cybersecurity Law Rev.*, vol. 3, no. 2, pp. 437–465, 2022.

[59] F. Aldauiji, O. Batarfi, and M. Bayousef, "Utilizing cyber threat hunting techniques to find ransomware attacks: A survey of the state of the art," *IEEE Access*, vol. 10, pp. 61695–61706, 2022.

[60] H. Pu, L. He, P. Cheng, M. Sun, and J. Chen, "Security of industrial robots: Vulnerabilities, attacks, and mitigations," *IEEE Netw.*, vol. 37, no. 1, pp. 111–117, Jan./Feb. 2023.

[61] I. B. A. Ouahab, M. Bouhorma, L. E. Aachak, and A. A. Boudhir, "Towards a new cyberdefense generation: Proposition of an intelligent cybersecurity framework for malware attacks," *Adv. Comput. Sci. Commun.*, vol. 15, no. 8, pp. 1026–1042, 2022.

[62] J. Martínez and J. M. Durán, "Software supply chain attacks, a threat to global cybersecurity: SolarWinds' case study," *Int. J. Safety Security Eng.*, vol. 11, no. 5, pp. 537–545, 2021.

[63] S. Amjad, S. Abbas, Z. Abubaker, M. H. Alsharif, A. Jahid, and N. Javaid, "Blockchain based authentication and cluster head selection using DDR-LEACH in Internet of Sensor Things," *Sensors*, vol. 22, no. 5, p. 1972, 2022.

[64] S. Berger, O. Bürger, and M. Röglinger, "Attacks on the Industrial Internet of Things—Development of a multi-layer taxonomy," *Comput. Security*, vol. 93, Jun. 2020, Art. no. 101790.

[65] N. Kadham and K. S. Ravi, "A lightweight one time password (OTP) based smart learning in Internet of Things," *Int. J. Eng. Technol.*, vol. 7, no. 2, pp. 480–483, 2018.

[66] H. A. Tarish, "Enhanced IoT Wi-Fi protocol standard's security using secure remote password," *Periodicals Eng. Nat. Sci.*, vol. 10, no. 1, pp. 632–644, 2022.

[67] C. Faircloth, G. Hartzell, N. Callahan, and S. Bhunia, "A study on brute force attack on T-mobile leading to SIM-Hijacking and identity-theft," in *Proc. IEEE World AI IoT Congr. (AIIoT)*, 2022, pp. 501–507.

[68] P. Ferrari et al., "Turning old into new: Adding LoRaWAN connectivity to PLC in brownfield installations," in *Proc. IEEE Int. Workshop Metrol. Ind. 4.0 IoT (MetroInd4.0 IoT)*, 2021, pp. 665–670.

[69] M. Safwat, A. Elgammal, E. G. AbdAllah, and M. A. Azer, "Survey and taxonomy of information-centric vehicular networking security attacks," *Ad Hoc Netw.*, vol. 124, Jan. 2022, Art. no. 102696.

[70] M. A. Ferrag, L. Maglaras, A. Derhab, and H. Janicke, "Authentication schemes for smart mobile devices: Threat models, countermeasures, and open research issues," *Telecommun. Syst.*, vol. 73, pp. 317–348, Sep. 2020.

[71] L. Basyoni, N. Fetais, A. Erbad, A. Mohamed, and M. Guizani, "Traffic analysis attacks on Tor: A survey," in *Proc. IEEE Int. Conf. Inf. IoT Enabling Technol. (ICIoT)*, 2020, pp. 183–188.

[72] R. C. Rao, K. M. Lakshmi, C. Raja, P. B. S. Varma, G. R. K. Rao, and A. Patibandla, "Real-time implementation and testing of VoIP vocoders with asterisk PBX using wireshark packet analyzer," *J. Interconnection Netw.*, vol. 22, no. S1, 2022, Art. no. 2141030.

[73] A. F. Baig and S. Eskeland, "Security, privacy, and usability in continuous authentication: A survey," *Sensors*, vol. 21, no. 17, p. 5967, 2021.

[74] U. Ghugar and J. Pradhan, "Survey of wormhole attack in wireless sensor networks," *Comput. Sci. Inf. Technol.*, vol. 2, no. 1, pp. 33–42, 2021.

[75] A. A. Mahamune and M. Chandane, "TCP/IP layerwise taxonomy of attacks and defence mechanisms in mobile ad hoc networks," *J. Inst Eng. B*, vol. 103, no. 1, pp. 273–291, 2022.

[76] A. Mehta, J. K. Sandhu, M. Pundir, R. Kaur, and L. Sapra, "Sinkhole attack detection in wireless sensor networks," in *Proc. Data Anal. Manag. (ICDAM)*, vol. 2, 2022, pp. 85–94.

[77] S. K. Tetarave, S. Tripathy, E. Kalaimannan, C. John, and A. Srivastava, "A routing table poisoning model for peer-to-peer (P2P) botnets," *IEEE Access*, vol. 7, pp. 67983–67995, 2019.

[78] Y. Al-Hadhrami and F. K. Hussain, "DDoS attacks in IoT networks: A comprehensive systematic literature review," *World Wide Web*, vol. 24, no. 3, pp. 971–1001, 2021.

[79] M. Snehi and A. Bhandari, "Vulnerability retrospection of security solutions for software-defined cyber-physical system against DDoS and IoT-DDoS attacks," *Comput. Sci. Rev.*, vol. 40, May 2021, Art. no. 100371.

[80] J. F. Balarezo, S. Wang, K. G. Chavez, A. Al-Hourani, and S. Kandeepan, "A survey on DoS/DDoS attacks mathematical modelling for traditional, SDN and virtual networks," *Eng. Sci. Technol. Int. J.*, vol. 31, Jul. 2022, Art. no. 101065.

[81] A. Adeel et al., "A multi-attack resilient lightweight IoT authentication scheme," *Trans. Emerg. Telecommun. Technol.*, vol. 33, no. 3, 2022, Art. no. e3676.

[82] L. Wang, X. Cao, H. Zhang, C. Sun, and W. X. Zheng, "Transmission scheduling for privacy-optimal encryption against eavesdropping attacks on remote state estimation," *Automatica*, vol. 137, Mar. 2022, Art. no. 110145.

[83] J. H. Anajemba, C. Iwendi, I. Razzak, J. A. Ansere, and I. M. Okpalaoguchi, "A counter-eavesdropping technique for optimized privacy of wireless Industrial IoT communications," *IEEE Trans. Ind. Informat.*, vol. 18, no. 9, pp. 6445–6454, Sep. 2022.

[84] X. Zhang, J. Hamm, M. K. Reiter, and Y. Zhang, "Defeating traffic analysis via differential privacy: A case study on streaming traffic," *Int. J. Inf. Security*, vol. 21, no. 3, pp. 689–706, 2022.

[85] K. Demertzis, K. Tsiknas, D. Takezis, C. Skianis, and L. Iliadis, "DarkNet traffic big-data analysis and network management for real-time automating of the malicious intent detection process by a weight agnostic neural networks framework," *Electronics*, vol. 10, no. 7, p. 781, 2021.

[86] F. Carrera, V. Dentamaro, S. Galantucci, A. Iannacone, D. Impedovo, and G. Pirlo, "Combining unsupervised approaches for near real-time network traffic anomaly detection," *Appl. Sci.*, vol. 12, no. 3, p. 1759, 2022.

[87] N. Sharma, N. Chauhan, N. Chand, and L. K. Awasthi, "Secure authentication and session key management scheme for Internet of Vehicles," *Trans. Emerg. Telecommun. Technol.*, vol. 33, no. 5, 2022, Art. no. e4451.

[88] W. Xu, J. Kurths, G. Wen, and X. Yu, "Resilient event-triggered control strategies for second-order consensus," *IEEE Trans. Autom. Control*, vol. 67, no. 8, pp. 4226–4233, Aug. 2021.

[89] N. Kumari and A. Mohapatra, "A comprehensive and critical analysis of TLS 1.3," *J. Inf. Optim. Sci.*, vol. 43, no. 4, pp. 689–703, 2022.

[90] X. Zhang, Q. Cheng, and Y. Li, "LaTLS: A lattice-based TLS proxy protocol," *Chin. J. Electron.*, vol. 31, no. 2, pp. 313–321, 2022.

[91] B. Reddy et al., "The AODV routing protocol with built-in security to counter blackhole attack in MANET," *Mater. Today Proc.*, vol. 50, no. 5, pp. 1152–1158, 2022.

[92] Z. Teng, C. Du, M. Li, H. Zhang, and W. Zhu, "A wormhole attack detection algorithm integrated with the node trust optimization model in WSNs," *IEEE Sensors J.*, vol. 22, no. 7, pp. 7361–7370, Apr. 2022.

[93] S. Singh and H. S. Saini, "Intelligent ad-hoc-on demand multipath distance vector for wormhole attack in clustered WSN," *Wireless Pers. Commun.*, vol. 122, no. 2, pp. 1305–1327, 2022.

[94] T. Thiyagu, S. Krishnaveni, and R. Arthi, "Deep learning approach for RPL wormhole attack," in *Proc. Intell. Data Commun. Technol. Internet Things (ICICI)*, 2022, pp. 321–330.

[95] S. A. Bhosale and S. Sonavane, "Wormhole attack detection system for IoT network: A hybrid approach," *Wireless Pers. Commun.*, vol. 124, no. 2, pp. 1081–1108, 2022.

[96] G. Soni, K. Chandravanshi, M. K. Jhariya, and A. Rajput, "An IPS approach to secure V-RSU communication from blackhole and wormhole attacks in VANET," in *Proc. Contemp. Issues Commun. Cloud Big Data Anal. (CCB)*, 2022, pp. 57–65.

[97] A. K. Sangaiah, A. Javadpour, F. Ja'fari, P. Pinto, H. Ahmadi, and W. Zhang, "CL-MLSP: The design of a detection mechanism for sinkhole attacks in smart cities," *Microprocess. Microsyst.*, vol. 90, Apr. 2022, Art. no. 104504.

[98] D. Bhattacharya, N. S. H. Karthick, P. Suresh, and N. Bhalaji, "DetecSec: A framework to detect and mitigate ARP cache poisoning attacks," in *Proc. Evol. Comput. Mobile Sustain. Netw. (ICECMSN)*, 2022, pp. 997–1007.

[99] G. Jethava and U. P. Rao, "User behavior-based and graph-based hybrid approach for detection of sybil attack in online social networks," *Comput. Elect. Eng.*, vol. 99, Apr. 2022, Art. no. 107753.

[100] Y.-S. Yang, S.-H. Lee, W.-C. Chen, C.-S. Yang, Y.-M. Huang, and T.-W. Hou, "Securing SCADA energy management system under DDoS attacks using token verification approach," *Appl. Sci.*, vol. 12, no. 1, p. 530, 2022.

[101] T. G. Zewdie and A. Girma, "An evaluation framework for machine learning methods in detection of DoS and DDoS intrusion," in *Proc. IEEE Int. Conf. Artif. Intell. Inf. Commun. ICAIIC)*, 2022, pp. 115–121.

[102] S. Ghayyad, S. Du, and A. Kurien, "The flaws of Internet of Things (IoT) intrusion detection and prevention schemes," *Int. J. Sensor Netw.*, vol. 38, no. 1, pp. 25–36, 2022.

[103] A. Mallik, A. Ahsan, M. Shahadat, and J. Tsou, "Understanding man-in-the-middle-attack through survey of literature," *Indonesian J. Comput. Eng. Design*, vol. 1, no. 1, pp. 44–56, 2019.

[104] A. Bhardwaj, V. Mangat, R. Vig, S. Halder, and M. Conti, "Distributed denial of service attacks in cloud: State-of-the-art of scientific and commercial solutions," *Comput. Sci. Rev.*, vol. 39, Feb. 2021, Art. no. 100332.

[105] A. K. Jain and B. Gupta, "A survey of phishing attack techniques, defence mechanisms and open research challenges," *Enterprise Inf. Syst.*, vol. 16, no. 4, pp. 527–565, 2022.

[106] A. F. Ghazali et al., "A survey of malware risk detection techniques in cloud," *Turkish J. Comput. Math. Educ.*, vol. 12, no. 3, pp. 868–876, 2021.

[107] S. Abdulkarim, S. K. Ahmad, and F. Binord, "A review of faults attack on symmetric key cipher and appropriate counter measures," *OIRT J. Inf. Technol.*, vol. 2, no. 1, pp. 1–7, 2022.

[108] H. Alsulami, "Implementation analysis of reliable unmanned aerial vehicles models for security against cyber-crimes: Attacks, tracebacks, forensics and solutions," *Comput. Elect. Eng.*, vol. 100, May 2022, Art. no. 107870.

[109] M. M. Ahsan, I. Ali, M. Y. I. B. Idris, M. Imran, and M. Shoaib, "Countering statistical attacks in cloud-based searchable encryption," *Int. J. Parallel Program.*, vol. 48, pp. 470–495, Jun. 2020.

[110] M. Abu-Alhaija, N. M. Turab, and A. Hamza, "Extensive study of cloud computing technologies, threats and solutions prospective," *Comput. Syst. Sci. Eng.*, vol. 41, no. 1, pp. 225–240, 2022.

[111] H. Tabrizchi and M. K. Rafsanjani, "A survey on security challenges in cloud computing: Issues, threats, and solutions," *J. Supercomput.*, vol. 76, no. 12, pp. 9493–9532, 2020.

[112] H. K. Bella and S. Vasundra, "A study of security threats and attacks in cloud computing," in *Proc. 4th Int. Conf. Smart Syst. Inventive Technol. (ICSSIT)*, 2022, pp. 658–666.

[113] F. Aliyu, T. Sheltami, and E. M. Shakshuki, "A detection and prevention technique for man in the middle attack in fog computing," *Procedia Comput. Sci.*, vol. 141, pp. 24–31, Jan. 2018.

[114] B. Gupta, P. Chaudhary, X. Chang, and N. Nedjah, "Smart defense against distributed denial of service attack in IoT networks using supervised learning classifiers," *Comput. Elect. Eng.*, vol. 98, Mar. 2022, Art. no. 107726.

[115] S. Kautish, A. Reyana, and A. Vidyarthi, "SDMTA: Attack detection and mitigation mechanism for DDoS vulnerabilities in hybrid cloud environment," *IEEE Trans. Ind. Informat.*, vol. 18, no. 9, pp. 6455–6463, Sep. 2022.

[116] A. Huszti, S. Kovács, and N. Oláh, "Scalable, password-based and threshold authentication for smart homes," *Int. J. Inf. Security*, vol. 21, no. 4, pp. 707–723, 2022.

[117] S. Ciolino, S. Parkin, and P. Dunphy, "Of two minds about {two-factor}: Understanding everyday {FIDO}{U2F} usability through device comparison and experience sampling," in *Proc. 15th Symp. Usable Privacy Security (SOUPS)*, 2019, pp. 339–356.

[118] K. Hageman, E. Kidmose, R. R. Hansen, and J. M. Pedersen, "Can a TLS certificate be phishy?" in *Proc. 18th Int. Conf. Security Cryptography (SECRYPT)*, 2021, pp. 38–49.

[119] J. Yu, Y. He, Q. Yan, and X. Kang, "SpecView: Malware spectrum visualization framework with singular spectrum transformation," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 5093–5107, 2021.

[120] R. Kumar, K. Sethi, N. Prajapati, R. R. Rout, and P. Bera, "Machine learning based malware detection in cloud environment using clustering approach," in *Proc. 11th Int. Conf. Comput. Commun. Netw. Technol. (ICCCNT)*, 2020, pp. 1–7.

[121] A. Bhardwaj, M. D. Alshehri, K. Kaushik, H. J. Alyamani, and M. Kumar, "Secure framework against cyber attacks on cyber-physical robotic systems," *J. Electron. Imag.*, vol. 31, no. 6, 2022, Art. no. 61802.

[122] S. Banani, S. Thiemjarus, K. Wongthavarawat, and N. Ounanong, "A dynamic light-weight symmetric encryption algorithm for secure data transmission via BLE beacons," *J. Sensor Actuator Netw.*, vol. 11, no. 1, p. 2, 2021.

[123] M. P. Priyanka et al., "A comparative review between modern encryption algorithms viz. DES, AES, and RSA," in *Proc. Int. Conf. Comput. Intell. Sustain. Eng. Solutions (CISES)*, 2022, pp. 295–300.

[124] S. El Kafhali, I. El Mir, and M. Hanini, "Security threats, defense mechanisms, challenges, and future directions in cloud computing," *Arch. Comput. Methods Eng.*, vol. 29, no. 1, pp. 223–246, 2022.

[125] P. Chauhan, S. Ahmad, P. R. Khan, and N. A. Khan, "Investigating the IoT security and privacy challenges: Summary and recommendations," *EAI Endorsed Trans. Cloud Syst.*, vol. 7, no. 22, p. e5, 2022.

[126] A. Salman, M. S. Khan, S. Idrees, F. Akram, M. Junaid, and A. L. Malik, "File integrity checkers: Functionality, attacks, and protection," in *Proc. 2nd Int. Conf. Digit. Futures Transformative Technol. (ICoDT2)*, 2022, pp. 1–6.

[127] D. An, F. Zhang, Q. Yang, and C. Zhang, "Data integrity attack in dynamic state estimation of smart grid: Attack model and countermeasures," *IEEE Trans. Autom. Sci. Eng.*, vol. 19, no. 3, pp. 1631–1644, Jul. 2022.

[128] V. K. Singh, N. Bharathiraja, S. Arun, D. B. David, R. Krishnamoorthy, and R. Thiagarajan, "Secure shared data in the private cloud with an EA algorithm," in *Proc. 8th Int. Conf. Smart Struct. Syst. (ICSSS)*, 2022, pp. 1–6.

[129] F. Y. Joe and V. Selvarajah, "A study of SQL injection hacking techniques," in *Proc. 3rd Int. Conf. Integr. Intell. Comput. Commun. Security (ICIIC)*, 2021, pp. 531–539.

[130] B. Kumar, V. Sridhar, and K. Sudhindra, "Generic security risk profile of e-governance applications—A case study," in *Proc. Emerg. Res. Comput. Inf. Commun. Appl. (ERCICA)*, vol. 2, 2022, pp. 731–741.

[131] A. Rai et al., "SQL injection: Classification and prevention," in *Proc. 2nd Int. Conf. Intell. Eng. Manag. (ICIEM)*, 2021, pp. 367–372.

[132] A. Raich and V. Gadicha, "Overview of passive attacks in cloud environment," in *Proc. AIP Conf.*, vol. 2424, 2022, Art. no. 30004.

[133] C. Gupta, R. K. Singh, and A. K. Mohapatra, "A survey and classification of XML based attacks on Web applications," *Inf. Security J. Global Perspect.*, vol. 29, no. 4, pp. 183–198, 2020.

[134] S. Ibarra-Fiallos, J. B. Higuera, M. Intriago-Pazmiño, J. R. B. Higuera, J. A. S. Montalvo, and J. Cubo, "Effective filter for common injection attacks in online Web applications," *IEEE Access*, vol. 9, pp. 10378–10391, 2021.

[135] S. K. Lala, A. Kumar, and T. Subbulakshmi, "Secure Web development using OWASP guidelines," in *Proc. 5th Int. Conf. Intell. Comput. Control Syst. (ICICCS)*, 2021, pp. 323–332.

[136] P. Nagarjun and S. A. Shaik, "Cross-site scripting research: A review," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 4, p. 14, 2020.

[137] N. A. A. Talib, "Static analysis tools against cross-site scripting vulnerabilities in Web applications: An analysis," *J. Softw. Assess. Valuation*, vol. 17, no. 2, pp. 125–142, 2021.

[138] Y. Yu, Z. Chen, S. Gan, and X. Qin, "Research on the technologies of security analysis technologies on the embedded device firmware," *Chin. J. Comput.*, vol. 44, no. 5, pp. 859–881, 2021.

[139] S. S. Ensan, K. Nagarajan, M. N. I. Khan, and S. Ghosh, "SCARE: Side channel attack on in-memory computing for reverse engineering," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 29, no. 12, pp. 2040–2051, Dec. 2021.

[140] P. Pessl, D. Gruss, C. Maurice, M. Schwarz, and S. Mangard, "DRAMA: Exploiting DRAM addressing for cross-CPU attacks," in *Proc. 25th USENIX Security Symp. (USENIX Security)*, 2016, pp. 565–581.

[141] W. Jin, S. Ullah, D. Yoo, and H. Oh, "NPDHunter: Efficient null pointer dereference vulnerability detection in binary," *IEEE Access*, vol. 9, pp. 90153–90169, 2021.

[142] S. Alouneh, M. Kharbutli, and R. AlQurem, "A software approach for stack memory protection based on duplication and randomisation," *Int. J. Internet Technol. Secured Trans.*, vol. 6, no. 4, pp. 324–348, 2016.

[143] L. Duan et al., "Multiple-layer security threats on the Ethereum blockchain and their countermeasures," *Security Commun. Netw.*, vol. 2022, Feb. 2022, Art. no. 5307697.

[144] L. A. C. Ahakonye, C. I. Nwakanma, S. O. Ajakwe, J. M. Lee, and D.-S. Kim, "Countering DNS vulnerability to attacks using ensemble learning," in *Proc. IEEE Int. Conf. Artif. Intell. Inf. Commun. (ICAIIC)*, 2022, pp. 7–10.

[145] O. Bamasag, A. Alsaeedi, A. Munshi, D. Alghazzawi, S. Alshehri, and A. Jamjoom, "Real-time DDoS flood attack monitoring and detection (RT-AMD) model for cloud computing," *PeerJ Comput. Sci.*, vol. 7, p. e814, Jun. 2022.

[146] A. Dhanapal and P. Nithyanandam, "An OpenStack based cloud testbed framework for evaluating HTTP flooding attacks," *Wireless Netw.*, vol. 27, no. 8, pp. 5491–5501, 2021.

[147] A. Dhanapal and P. Nithyanandam, "The HTTP flooding attack detection to secure and safeguard online applications in the cloud," *Int. J. Inf. Syst. Model. Design*, vol. 10, no. 3, pp. 41–58, 2019.

[148] S. Gupta and B. B. Gupta, "Evaluation and monitoring of XSS defensive solutions: A survey, open research issues and future directions," *J. Ambient Intell. Humanized Comput.*, vol. 10, pp. 4377–4405, Nov. 2019.

[149] Y. Abdulmalik, "An improved SQL injection attack detection model using machine learning techniques," *Int. J. Innov. Comput.*, vol. 11, no. 1, pp. 53–57, 2021.

[150] K. Selvamani and A. Kannan, "A novel approach for prevention of SQL injection attacks using cryptography and access control policies," in *Proc. Int. Conf. Power Electron. Instrum. Eng.*, 2011, pp. 26–33.

[151] M. Alahmad, A. Alkandari, and N. Alawadhi, "Survey of OS command injection Web application vulnerability attack," *J. Eng. Sci. Technol.*, vol. 17, no. 1, pp. 75–84, 2022.

[152] M. Ivanova and A. Rozeva, "Detection of XSS attack and defense of REST Web service—Machine learning perspective," in *Proc. 5th Int. Conf. Mach. Learn. Soft Comput.*, 2021, pp. 22–28.

[153] H. Shahriar, H. Haddad, and P. Bulusu, "LDAP vulnerability detection in Web applications," *Int. J. Secure Softw. Eng.*, vol. 8, no. 4, pp. 31–50, 2017.

[154] F. M. M. Mokbal, W. Dan, A. Imran, L. Jiuchuan, F. Akhtar, and W. Xiaoxi, "MLPXSS: An integrated XSS-based attack detection scheme in Web applications using multilayer perceptron technique," *IEEE Access*, vol. 7, pp. 100567–100580, 2019.

[155] D. Tiganov, J. Cho, K. Ali, and J. Dolby, "SWAN: A static analysis framework for swift," in *Proc. 28th ACM Joint Meeting Eur. Softw. Eng. Conf. Symp. Found. Softw. Eng.*, 2020, pp. 1640–1644.

[156] S. Habchi, G. Hecht, R. Rouvoy, and N. Moha, "Code smells in IoS apps: How do they compare to Android?" in *Proc. IEEE/ACM 4th Int. Conf. Mobile Softw. Eng. Syst. (MOBILESoft)*, 2017, pp. 110–121.

[157] M. A. Marin, C. Carabas, R. Deaconescu, and N. Tăpus, "Proactive secure coding for iOS applications," in *Proc. IEEE 18th RoEduNet Conf. Netw. Educ. Res. (RoEduNet)*, 2019, pp. 1–5.

[158] M. Weninger, E. Gander, and H. Mössenböck, "Utilizing object reference graphs and garbage collection roots to detect memory leaks in offline memory monitoring," in *Proc. 15th Int. Conf. Manag. Lang. Runtimes*, 2018, pp. 1–13.

[159] A. F. Blanco, A. Bergel, and J. P. S. Alcocer, "Software visualizations to analyze memory consumption: A literature review," *ACM Comput. Surveys*, vol. 55, no. 1, pp. 1–34, 2022.

[160] Q. Zhang, Y. Zhao, W. Sun, C. Fang, Z. Wang, and L. Zhang, "Program repair: Automated vs. manual," 2022, *arXiv:2203.05166*.

[161] S. Black and Y. Kim, "An overview on detection and prevention of application layer DDoS attacks," in *Proc. IEEE 12th Annu. Comput. Commun. Workshop Conf. (CCWC)*, 2022, pp. 791–800.

[162] N. Lu, J. Zhang, X. Liu, W. Shi, and J. Ma, "STOP: A service oriented Internet purification against link flooding attacks," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 938–953, 2022.

[163] D. Nashat and S. Khairy, "Detecting HTTP flooding attacks based on uniform model," in *Proc. Int. Conf. Netw. Netw. Appl. (NaNA)*, 2021, pp. 94–98.

[164] S. Park, Y. Kim, H. Choi, Y. Kyung, and J. Park, "HTTP DDoS flooding attack mitigation in software-defined networking," *IEICE Trans. Inf. Syst.*, vol. 104, no. 9, pp. 1496–1499, 2021.

[165] A. R. Ghilen, A. M. Zahou, and W. A. B. Khalifa, "Quantum cryptography for the benefit of API keys safety," in *Proc. IEEE 9th Int. Conf. Sci. Electron. Technol. Inf. Telecommun. (SETIT)*, 2022, pp. 343–348.

[166] R. L. Kumar, F. Khan, S. Kadry, and S. Rho, "A survey on blockchain for Industrial Internet of Things," *Alexandria Eng. J.*, vol. 61, no. 8, pp. 6001–6022, 2022.

[167] Y. Xie, Y. Li, and Y. Ma, "Data privacy security mechanism of Industrial Internet of Things based on block chain," *Appl. Sci.*, vol. 12, no. 14, p. 6859, 2022.

[168] B. Ernest and J. Shiguang, "Privacy enhancement scheme (PES) in a blockchain-edge computing environment," *IEEE Access*, vol. 8, pp. 25863–25876, 2020.

[169] Y. Wang, T. Che, X. Zhao, T. Zhou, K. Zhang, and X. Hu, "A blockchain-based privacy information security sharing scheme in Industrial Internet of Things," *Sensors*, vol. 22, no. 9, p. 3426, 2022.

[170] C.-L. Chen, J. Yang, W.-J. Tsaur, W. Weng, C.-M. Wu, and X. Wei, "Enterprise data sharing with privacy-preserved based on hyperledger fabric blockchain in IIoT's application," *Sensors*, vol. 22, no. 3, p. 1146, 2022.

[171] T. Feng, P. Yang, C. Liu, J. Fang, and R. Ma, "Blockchain data privacy protection and sharing scheme based on zero-knowledge proof," *Wireless Commun. Mobile Comput.*, vol. 2022, pp. 1–11, Feb. 2022.

[172] H. Wang, Y. Xiao, Y. Feng, Q. Qian, Y. Li, and X. Fu, "Cloud-assisted privacy protection energy trading based on IBS and homomorphic encryption in IIoT," *Appl. Sci.*, vol. 12, no. 19, p. 9509, 2022.

[173] Z. Bao, D. He, M. K. Khan, M. Luo, and Q. Xie, "PBidm: Privacy-preserving blockchain-based identity management system for Industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 19, no. 2, pp. 1524–1534, Feb. 2023.

[174] Y. Rahulamathavan, R. C.-W. Phan, M. Rajarajan, S. Misra, and A. Kondoz, "Privacy-preserving blockchain based IoT ecosystem using attribute-based encryption," in *Proc. IEEE Int. Conf. Adv. Netw. Telecommun. Syst. (ANTS)*, 2017, pp. 1–6.

[175] S. Jiang et al., "Privacy-preserving and efficient multi-keyword search over encrypted data on blockchain," in *Proc. IEEE Int. Conf. Blockchain (Blockchain)*, 2019, pp. 405–410.

[176] J. Feng, L. T. Yang, R. Zhang, and B. S. Gavuna, "Privacy-preserving tucker train decomposition over blockchain-based encrypted Industrial IoT data," *IEEE Trans. Ind. Informat.*, vol. 17, no. 7, pp. 4904–4913, Jul. 2021.

[177] S. Zhao, S. Li, and Y. Yao, "Blockchain enabled Industrial Internet of Things technology," *IEEE Trans. Comput. Social Syst.*, vol. 6, no. 6, pp. 1442–1453, Dec. 2019.

[178] K. Zhao, S. Tang, B. Zhao, and Y. Wu, "Dynamic and privacy-preserving reputation management for blockchain-based mobile crowdsensing," *IEEE Access*, vol. 7, pp. 74694–74710, 2019.

[179] J. Lee et al., "PUFTAP-IoT: PUF-based three-factor authentication protocol in IoT environment focused on sensing devices," *Sensors*, vol. 22, no. 18, p. 7075, 2022.

[180] F. A. Aponte-Novoa, A. L. S. Orozco, R. Villanueva-Polanco, and P. Wightman, "The 51% attack on blockchains: A mining behavior study," *IEEE Access*, vol. 9, pp. 140549–140564, 2021.

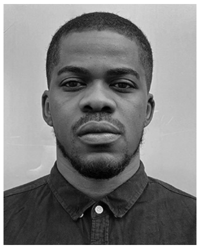
[181] X. Yan, K. Yan, M. U. Rehman, and S. Ullah, "Impersonation attack detection in mobile edge computing by levering SARSA technique in physical layer security," *Appl. Sci.*, vol. 12, no. 20, 2022, Art. no. 10225.

[182] P. J. Sun, "Research on selection method of privacy parameter," *Security Commun. Netw.*, vol. 2020, pp. 1–12, Oct. 2020.

[183] H. Liu, T. Gu, Y. Liu, J. Song, and Z. Zeng, "Fault-tolerant privacy-preserving data aggregation for smart grid," *Wireless Commun. Mobile Comput.*, vol. 2020, pp. 1–10, Sep. 2020.

[184] T.-Y. Wu, L. Yang, Z. Lee, C.-M. Chen, J.-S. Pan, and S. H. Islam, "Improved ECC-based three-factor multiserver authentication scheme," *Security Commun. Netw.*, vol. 2021, pp. 1–14, Jan. 2021.

[185] H. Shi, N. Yang, H. Tang, and X. Yang, "aSGD: Stochastic gradient descent with adaptive batch size for every parameter," *Mathematics*, vol. 10, no. 6, p. 863, 2022.



Anselme Herman Eyeleko received the B.S. degree in communication engineering and the M.S. degree in communication and information systems from Lanzhou University of Technology, Lanzhou, China, in 2016 and 2021, respectively, where he is currently pursuing the Ph.D. degree with the School of Computer and Communication.

His research interests include network security, data access control, Industrial Internet security, and data publishing privacy protection technology.

Tao Feng received the Ph.D. degree from Xidian University, Xi’an, China, in 2008.

He is currently a Professor and a Ph.D. Supervisor with Lanzhou University of Technology, Lanzhou, China. His research focuses on network and information security, Blockchain, and Industrial Internet security.