algorithms

MDPI

Article

Security Audit of a Blockchain-Based Industrial Application Platform

Jan Stodt [1,†], Daniel Schönle [1,†], Christoph Reich [1,†], Fatemeh Ghovanlooy Ghajar [2,†], Dominik Welte [2,*,†] and Axel Sikora [2,†]

[1] Institute for Data Science, Hochschule Furtwangen University, Cloud Computing and IT-Security (IDACUS), 78120 Furtwangen im Schwarzwald, Germany; jan.stodt@hs-furtwangen.de (J.S.); daniel.schoenle@hs-furtwangen.de (D.S.); christoph.reich@hs-furtwangen.de (C.R.)

[2] Institute of Reliable Embedded Systems and Communication Electronics (ivESK) , Offenburg University of Applied Sciences, 77652 Offenburg, Germany; fatemeh.ghovanlooy@hs-offenburg.de (F.G.G.); axel.sikora@hs-offenburg.de (A.S.)

* Correspondence: dominik.welte@hs-offenburg.de

† These authors contributed equally to this work.

Abstract: In recent years, both the Internet of Things (IoT) and blockchain technologies have been highly influential and revolutionary. IoT enables companies to embrace Industry 4.0, the Fourth Industrial Revolution, which benefits from communication and connectivity to reduce cost and to increase productivity through sensor-based autonomy. These automated systems can be further refined with smart contracts that are executed within a blockchain, thereby increasing transparency through continuous and indisputable logging. Ideally, the level of security for these IoT devices shall be very high, as they are specifically designed for this autonomous and networked environment. This paper discusses a use case of a company with legacy devices that wants to benefit from the features and functionality of blockchain technology. In particular, the implications of retrofit solutions are analyzed. The use of the BISS:4.0 platform is proposed as the underlying infrastructure. BISS:4.0 is intended to integrate the blockchain technologies into existing enterprise environments. Furthermore, a security analysis of IoT and blockchain present attacks and countermeasures are presented that are identified and applied to the mentioned use case.

Keywords: blockchain; distributed ledger; legacy machines; maintenance; shop floor; security

Citation: Stodt, J.; Schönle, D.; Reich, C.; Ghovanlooy Ghajar, F.; Welte, D.; Sikora, A. Security Audit of a Blockchain-Based Industrial Application Platform. *Algorithms* **2021**, *14*, 121. https://doi.org/10.3390/a14040121

Academic Editor: Tsan-Ming Choi (Jason)

Received: 26 February 2021
Accepted: 7 April 2021
Published: 10 April 2021

1. Introduction

In industrial environments, there are a variety of digital technologies that can be categorized as "Digital product innovation", "Cyber-physical systems", "Smart objects" and "Smart, connected products" [1]. In industrial environments, "industrial control systems" or "process control systems" include automatic and semiautomatic monitoring, condition assessment, maintenance, management, and data collection [2]. Equipment operation and maintenance are major tasks in industrial production systems. An intelligent predictive maintenance system is considered a subcategory of Industry 4.0 [3]. The safety of personnel and equipment has to be provided as well, especially as production becomes increasingly automated. In the Fourth Industrial Revolution, sensors are used to detect abnormal conditions on the factory floor to protect machines and people [4]. With the increasing number of different parties forced to work together, it is difficult to establish a trusted partnership. Blockchain technology presents one possible solution to alleviate this problem. However, serious security concerns are emerging due to increased connectivity (e.g., IoT networks) and control outsourcing [5,6]. Researchers have identified numerous potential attacks in the Industry 4.0 environment [7]. There are many general attacks on IT infrastructures in the literature, and most of them fall into four basic classes: physical, network, software, and data attacks.

- Physical attacks are carried out by physically accessing the manufacturing environment. Attackers can change the location of measurement devices (e.g., sensors), destroy them, or disrupt power electronics [8]. Attacks on information technology include the following: tampering [9], sleep denial attack [10], and Permanent Denial of Service (PDoS) [11].
- Network attacks are carried out by manipulating the connections between different devices. In the age of the Internet of Things, devices and connections can be compromised from anywhere in the world. The most common forms of network attacks are traffic analysis attacks [9], routing information attacks [9], selective forwarding attacks [12], Sybil attacks [12], replay attacks [13], and DoS attacks [14].
- Software attacks are carried out by an attacker who manipulates software or exploits system vulnerabilities through malicious code injection [10]: viruses, worms, Trojan horses, spyware, adware [9], and malware [13].
- Data and encryption security are one of the main concerns in the industrial sector. Therefore, cloud-based offerings are viewed with skepticism in terms of security. The cloud can help improve security in cases where appropriate audit mechanisms are provided and firmware and software update procedures are managed. The main attacks on data attacks in this environment to avoid data inconsistency [15] are discussed. Attacks on encryption include side-channel attacks [16].
- Attacks on certificates and user accounts result in unauthorized access [11] and stolen identity [17].

The goal is to further narrow this categorization with respect to the industrial environment and a system architecture based on a distributed ledger.

1.1. Paper Organization

The paper is structured as follows: In Section 2, the use case of the BISS:4.0 platform is presented and described. In Section 3, attackers and their possible entry points into the architecture through various attacks are identified. The identified attacks are then categorized for clarity. In Section 4, the identified and categorized attacks are applied to the use case and applicable countermeasures are identified. In Section 5, the identified and categorized attacks are applied to real-world examples to demonstrate countermeasures and benefits of the blockchain. Finally, Section 6 summarizes the key aspects of the paper.

1.2. Contribution

This paper has the following contributions: it primarily serves as a guide for companies looking to integrate blockchain into their production environment. Therefore, retrofit solutions in the area of sensor-based data acquisition are examined regarding their security. Attacks on these solutions are shown, and suitable countermeasures using blockchain technology are presented. Furthermore, existing security risks of legacy machines are presented and applicable countermeasures using blockchain technology are shown. To give companies a sense of how attacks on their environment can be mitigated or even prevented using blockchain, various attack scenarios are described.

2. BISS:4.0 Architecture

In this section, the use case of the blockchain based Industry 4.0 platform BISS:4.0 platform (BISS:4.0 stands for "blockchain in the switching cabinet" and the fourth industrial revolution.) is described and aims to improve manual tasks, such as maintenance, supported by smart contracts in a blockchain framework and combined with sensor-based data collection; for a detailed description of the BISS:4.0 platform, see [18]. The use of blockchain technology fits the industrial use case due to the ability to handle a multitude of different stakeholders, which do not trust each other. The reason for choosing a private and permissioned blockchain instead of a public one is, that, even though the stakeholders do not trust each other, they, most likely, are already working together and know each other quite well. The blockchain then provides an immutable data storage that

is very useful in guaranteeing traceability. An overview of the BISS:4.0 architecture, its components, and connectors to industrial environments is shown in Figure 1, with the first implementation focusing on the vendor part. There is a strict separation between the components of the blockchain framework (Hyperledger Fabric, or Fabric for short (Fabric is a private and permissioned open source blockchain framework. For the basic architecture, see [19].)) and the surrounding systems. The system uses a combination of existing open source components and homegrown components described below. The Fabric nodes of the blockchain ledger in the middle of Figure 1 are accessed by the various stakeholders taking part in the blockchain network. An auxiliary database is connected to the blockchain to store large amounts of data (e.g., images and measurement data). A Sensor Software Development Kit (SDK) has access to the sensors of the machines and forwards the data to a Fabric peer. These hardware sensors are connected either via I^2C or via an OPC UA server. The condition of the machines is monitored by a condition monitoring service that uses either the Sensor SDK to read data directly from the sensors, the auxiliary database, or the blockchain depending on the data needed to monitor the devices. User interaction is implemented via a mobile application, a web frontend, or a command line interface (CLI). The mobile application and web frontend access the blockchain via the backend as it interacts with the Fabric SDK. Administrative processes, such as setup and user management, are handled through the CLI. The machine manufacturer (OEM) has the function to create and check the maintenance plans. A maintenance plan service manages the maintenance plans, and with the maintenance support adapter, the schemas are checked for conformity with the corresponding maintenance plan. The maintenance service provider then performs the maintenance task. They use the mobile application and the web front end with correspondingly restricted access rights. They can access the blockchain via the manufacturer node (if the manufacturer operates one). For their own applications (e.g., controlling), a Controlling SDK is used. They can also use their own Fabric infrastructure to access the network. An attacker could attack the system using the entry points described in Section 3.1. Figure 2 provides an overview of an example maintenance scenario, as described earlier with some possible attacks (the different attacks, shown as dashed arrows, are categorized in Figure 3; see Table 1 for the full list).

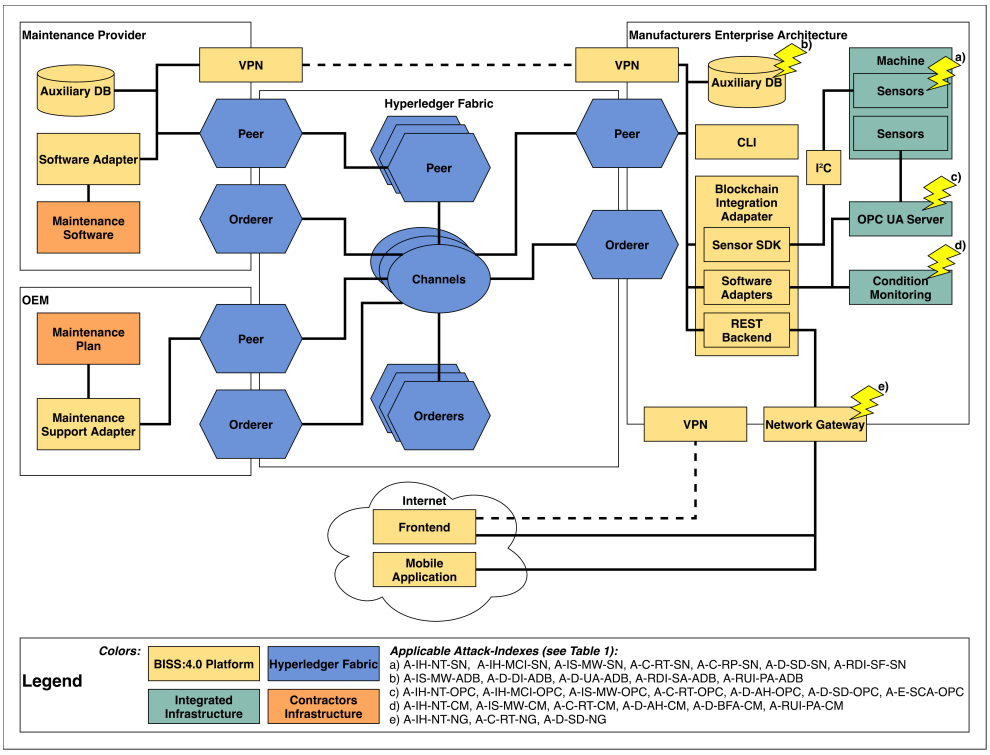

Figure 1. The BISS:4.0 platform with selected attacks (Indexes are taken from Table 1).

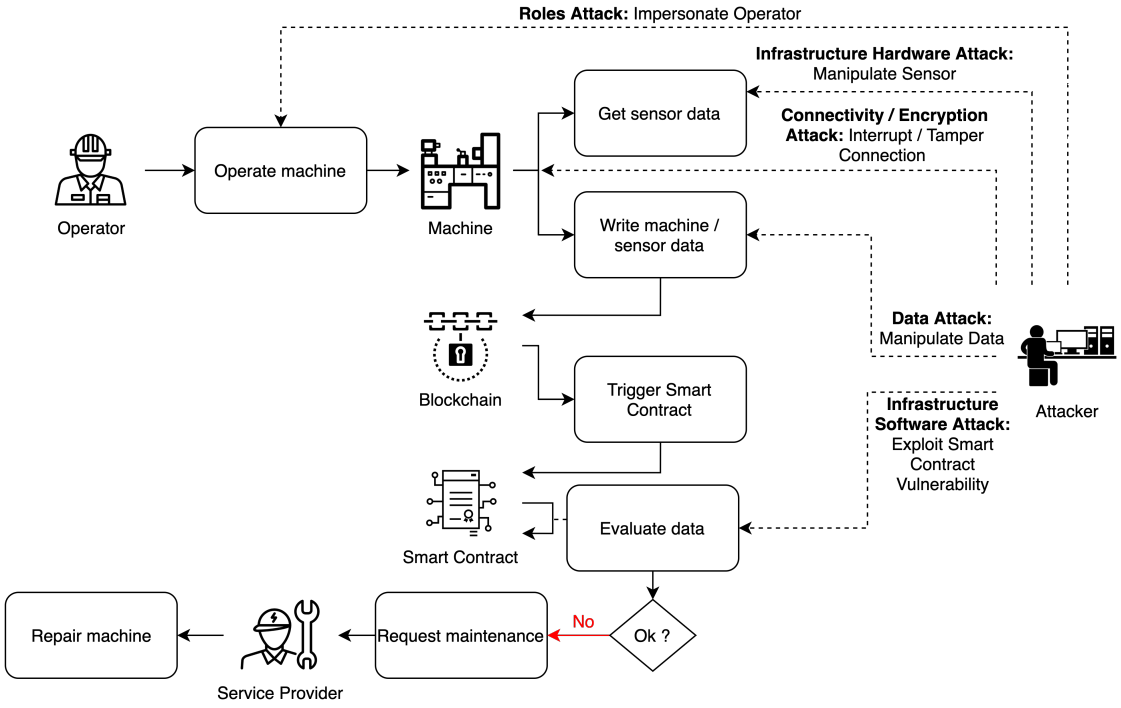

Figure 2. Diagram with attack possibilities (taken from the second column in Table 1).

3. Attacker Goals, Resources, and Capabilities

This section provides an overview of the attackers' goals, resources, and capabilities in executing an attack. Attackers could be hired by a competitor, be financially motivated, or simply want to damage the company's public image [20]. In addition, attackers could be interested in knowledge, such as process logic, which is usually fully protected intellectual property. A remote attacker without access to the factory network could attempt to create a malicious program for the industrial robot and execute it through an internal attack or other software-based attack on the robot. If an attacker knows that the system or factory integrator is using a specific production software, they will likely target the software itself or a third-party extension or library for that software.

3.1. Entry Points

This section provides a security-oriented overview of an Industry 4.0 architecture, from which attack levels and a number of entry points for an attacker are extracted. By highlighting the physical network environment, it is possible to directly see the different parts of the system architecture and their boundaries. Figure 1 shows the architecture of the BISS:4.0 platform, with yellow lightning bolts indicating endpoints that can be used as entry points for attacks. The focus is on examining the entry points located at the boundaries and edges of each part/system of the architecture rather than the issues within those parts/systems. In particular, the security of the blockchain framework itself is therefore not examined. An example of a detailed security analysis for a blockchain network can be seen in [21].

- *Engineering workstation:* the workstation is used to develop and access the various applications. Therefore, a trust relationship must exist between each workstation (and the software used on it) and the rest of the system. Although it is not assumed that the developer is malicious, their computer or a library that they use can be compromised. Since it is very likely that a developer uses third-party libraries [22], the software dependency chain can become very complex and impossible to fully verify, creating a remote attack surface.
- *Malicious devices:* additional devices connected to the network are becoming more popular, greatly increasing the attack surface. The miniaturization of electronic com-

ponents makes it possible for hardware implants to be as small as the metal part of a USB flash drive [23]. In fact, there have been cases where such devices have been used undetected as entry points to break into critical facilities.

- *Mobile application:* since mobile apps can interact with other apps on the same device, there is a risk that other apps installed on the device may not be trusted. Even though sandboxing of individual apps within mobile operating systems is well advanced, some apps still require permissions that undermine this separation. Another problem is that, although mobile devices today use apps from a closed ecosystem (walled garden/app stores), this does not mean that there are no malicious apps on the device [24]: Malicious apps are often removed from app stores because they spy on the user or try to escape the sandbox.
- *Legacy machine:* legacy machines are prevalent in an industrial setting, so it is critical that these legacy machines be integrated into the blockchain framework. Due to the nature of these machines, they are often connected via means that do not have built-in security concepts (e.g., simple serial connections) [25]. These security weaknesses can provide easy attack surfaces.
- *OPC UA server:* newer machines often provide connectivity via an integrated OPC UA server. This object-oriented industrial machine-to-machine protocol is a standard for industrial automation. While it has optional security features (TLS), it can be used without encryption and authentication [26]. OPC UA provides internal security mechanisms similar to TLS.
- *Sensor:* sensors in the architecture are often connected to machines via a simple fieldbus and are unsecured. This can create multiple attack surfaces, such as for data spoofing [27].
- *Auxiliary DB:* an auxiliary database is used to store information that is deemed too large or too frequent to be stored directly in the blockchain ledger. Potential attack surfaces are similar to ordinary NoSQL databases, e.g., data manipulation [28].
- *Condition monitoring:* the platform uses condition monitoring to monitor the various machines. In addition, the collected sensor data can be used, for example, to train a machine learning model. This model is then used to make predictions about the maintenance condition of the different machines (predictive maintenance). Possible attack surfaces are the sensor data and the data transmission [29].
- *Blockchain network:* the blockchain network itself has several entry points for attacks. First, it is possible to attack traffic between the various components of the network, and second, it is possible to exploit vulnerabilities in the smart contracts running within the network [30].
- *Network gateway:* the network gateway acts as a firewall between the organization's internal network and the external world [31]. External services can connect through gateways (such as Virtual Private Network (VPN) or public services) to communicate with the blockchain, for example.

3.2. Types of Attacks

In this section, attacks applicable to the described BISS:4.0 platform were investigated, categorized, described, and classified. While a variety of attacks exist on software and hardware, the following only discusses attacks that target the category of cyber-physical systems in the industrial environment [1]. The retrofitting of sensors in existing machines results in additional attack vectors from the category of IoT. As shown in Figure 3, attacks can be divided into five main categories: (1) data attacks, (2) communication channel attacks, (3) role attacks, (4) encryption attacks, and (5) infrastructure attacks (The attack categorization was partly inspired by [32], but IoT-specific parts were omitted and other parts were added or modified to fit the use case.). Attacks in the role category grant unauthorized access and can be prevented by using TLS encryption and authentication for general communication. Registering the identity of users in the blockchain prevents the following attacks: social engineering, phishing, spoofing credentials, and certificate

and spoofed node. In the mobile application area, only password attacks can occur in user identification and spoofing in device identification.

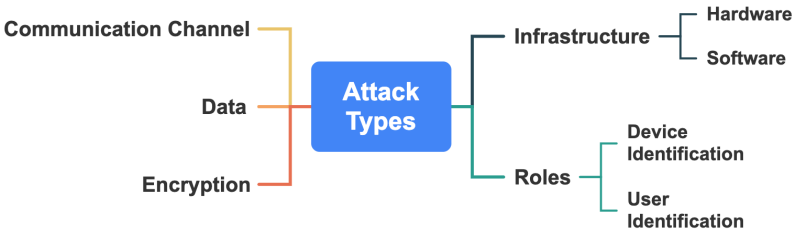

Figure 3. Types of attacks, adapted from [32].

The communication channel category includes attacks that manipulate or eavesdrop on traffic in the network part of the architecture. Most of these attacks can be mitigated by encrypting the traffic (by using TLS or a VPN). Active attacks such as DNS spoofing, handshake attacks, and signal manipulation are also difficult to perform when communicating only over encrypted channels. It can even be beneficial to combine these two solutions, for example, by tunneling all TLS traffic through an enterprise VPN. In the case of subsequent vulnerabilities in TLS itself (e.g., POODLE [33] and subsequently [34], which affected all TLS versions except TLS 1.2 with AEAD cipher suites) or in the case of a particular TLS implementation (e.g., Heartbleed [35]), there is an additional securely encrypted layer that guarantees that no data is exposed. There are several types of attacks that target the data flowing through the system: data exposure, data scavenging, and data tampering attacks are not effective because the system uses encrypted transport protocols, stores hashed data, and uses smart contracts to validate the data. In the category of encryption attacks, the risk of unknown vulnerabilities in cryptographic algorithms must be distinguished from attacks on a specific implementation, but both risks can be minimized by using known and widely used algorithms and implementations. Side channel attacks (e.g., meltdown) can be mitigated by upgrading the operating system or hardware. Attacks on infrastructure are divided into attacks on hardware and software. However, hardware attacks such as physical damage, social engineering, object replication attack, node manipulation, hardware Trojan, malicious code injection, node disruption, and sleep deprivation attacks on the proposed system are questionable because they require physical access to the system. It can be assumed that unauthorized physical access to the system is prevented by access controls. Software attacks such as phishing, malware, and spyware, on the other hand, can be carried out from a remote location.

3.3. Attack Classification

This section presents a classification of the different types of attacks on infrastructure consisting of the hardware and software parts, the communication channel used for data exchange, the data itself, the encryption used to protect the data from attacks, and roles in the area of device and user identification. A complete overview is provided by Table 1. Starting from the left, the individual types of attacks are broken down into more detailed subcategories the further you follow the table to the right. For example, the type of attack "infrastructure" breaks down into the subcategories hardware and software. These two subcategories, hardware and software, are each broken down further to show each individual attack type and the corresponding targets. As an example, the index A-IH-NT-CM represents an attack on a condition monitoring system that is assigned to the attack type infrastructures—hardware—node tampering. The other attacks mentioned are structured similarly. Each entry of the Table 1 is discussed in the following subsections.

Table 1. Overview of Attacks.

Index	Types of Attacks			Target
A-IH-NT-CM A-IH-NT-LM A-IH-NT-NG A-IH-NT-OPC A-IH-NT-SN	Infrastructure	Hardware	Node Tampering	Condition Monitoring Legacy Machine Network Gateway OPC UA Server Sensor
A-IH-MCI-LM A-IH-MCI-OPC A-IH-MCI-SN			Malicious Code Injection	Legacy Machine OPC UA Server Sensor
A-IH-SDA-LM			Sleep Denial Attack	Legacy Machine
A-IS-MW-ADB A-IS-MW-CM A-IS-MW-OPC A-IS-MW-SN		Software	Malware: Adware, Spyware, Trojan, Virus, and Worms	Auxiliary DB Condition Monitoring OPC UA Server Sensor
A-IS-TJ-CP A-IS-TJ-PB			Timejacking	Client to Peers Peers in Blockchain
A-C-RT-CM A-C-RT-LM A-C-RT-NG A-C-RT-OPC A-C-RT-SN	Communication Channel		Routing	Condition Monitoring Legacy Machine Network Gateway OPC UA Server Sensor
A-C-RP-CP A-C-RP-PB A-C-RP-SN			Replay	Client to Peers Peers in Blockchain Sensor
A-C-SF-CP			Selective Forwarding	Client to Peer
A-D-AH-CM A-D-AH-OPC	Data		Account Hijacking	Condition Monitoring OPC UA Server
A-D-BFA-CM A-D-BFA-CM			Brute Force Attack	Condition Monitoring OPC UA Server
A-D-DI-ADB			Data Inconsistency	Auxiliary DB
A-D-MSC-PB			Malicious Smart Contracts	Peers in Blockchain
A-D-SD-NG A-D-SD-OPC A-D-SD-SN			Sensor/User Data	Network Gateway OPC UA Server Sensor
A-D-UA-ADB			Unauthorized Access	Auxiliary DB
A-E-SCA-OPC A-E-SCA-PB	Encryption		Side Channel Attack	OPC UA Server Peers in Blockchain
A-RDI-SF-SN	Roles	Device Identification	Spoofing	Sensor
A-RDI-SA-ADB			Sybil Attack	Auxiliary DB
A-RUI-PA-ADB A-RUI-PA-CM A-RUI-PA-OPC		User Identification	Password Attack	Auxiliary DB Condition Monitoring OPC UA Server

3.3.1. Infrastructure Attacks

This section describes attacks on infrastructures and divides them into the subcategories of software and hardware and the attack groups.

Infrastructure—Hardware

- *Node tampering* refers to the act of physically modifying a device [36] or communication link. The business logic of an application is a target for parameter manipulation [37]. In this attack, the hidden or fixed file is used by programmers for a specific operation. Data manipulation is the practice of intentionally changing (destroying, manipulating, or editing) data.
- *Malicious code injection:* code injection is the exploitation of vulnerabilities in a computer system that leads to the execution of unwanted code in the system. The intruder can use injection attacks to change the behavior of the program in any direction [38].
- *Sleep denial attack:* the attacker keeps the battery-powered devices awake by feeding them with false inputs. This leads to the exhaustion of their batteries, causing them to shut down [39].

Infrastructure—Software

- *Malware: adware, spyware, Trojan, virus, and worms:* the IoT devices can be infected with malware that can spread to the cloud or data centers [40]. Using malware, an attacker can infect the system to manipulate data or steal information or even launch flooding attacks.
- *Timejacking:* an attacker typically changes the node's network time by including as many peers as possible and sending false timestamps on the network. This causes other peers to speed up and isolates the target from the network without interference from authentic nodes [41].

3.3.2. Communication Channel Attacks

This section describes attacks on communication channels between the blockchain and external systems such as state monitoring, machines, networks, and sensors.

- *Routing:* these are direct attacks in which the attacker forges or modifies routing information and causes problems through activities such as creating routing loops, sending error messages, and more [42]. Attackers can eavesdrop and disrupt the transmission channels. Even if the signals are encrypted, the attackers are able to analyze the signal streams and derive private information, such as the locations of sources or destinations. The attackers can also disrupt and even jam the wireless channels by sending noisy signals [43].
- *Replay:* an attacker can intercept a signed packet and send the packet back to the destination multiple times. This packet keeps the network busy, resulting in a DoS attack [44]. Advanced replay attacks hide the manipulated behavior of devices by sending old sensor data [45].
- *Selective forwarding:* in this attack, a malicious node selectively modifies messages, discards them, or forwards them to other nodes. Therefore, the information that reaches the destination is incomplete [46].

3.3.3. Data Attacks

This section describes attacks on data collected, processed, and stored in the systems used.

- *Account hijacking:* account hijacking aims to take over a user account through various attacks and to thereby gain access to protected data [47,48].
- *Brute force attack:* although there are many approaches to implementing brute force attacks, they all aim to guess a secret or force the system into a state where it can be more easily attacked by many attempts. This is also a problem in IoT networks [49].

- *Data inconsistency:* in IoT, a data integrity attack that results in inconsistency of data in transit or of data stored in a central database is called data inconsistency [50].
- *Malicious smart contracts:* smart contracts cannot handle unhandled code exceptions and procedure restructuring when validating transactions. The malicious smart contract is created and signed. The purpose is to execute the same expiration-invariant function over a contract before the original process is complete. By using the call function, the interaction with the main contract is called multiple times before its execution is complete [51].
- *Sensor/user data:* false data injection attacks refer to attackers sending false data with legitimate identities over the target network. Once the false data is accepted, IoT applications can return erroneous instructions or provide false services, affecting the reliability of IoT applications and networks [52].
- *Unauthorized access:* access control means granting access to authorized users and denying access to unauthorized users. With unauthorized access, malicious users can gain data ownership or access sensitive data [53].

3.3.4. Encryption Attacks

This section describes attacks on the encryption used to protect the data collected, processed, and stored in the systems used.

- *Side channel attack:* in this attack, the attacker collects the encryption keys by using timing, power, and fault attacks on the system's devices. Using these keys, they can encrypt and decrypt confidential data [54].

3.3.5. Roles Attacks

This section describes attacks on roles used within various systems and divides them into the subcategories of device identification and user identification as well as the attack groups.

Roles—Device Identification

- *Spoofing:* the attacker manipulates the records of a Domain Name System (DNS) server to redirect network traffic and to hide the source of the exploitation. As a result, a device spoofing attack can take control of sensors and manipulate data [55]. For example, with a device spoofing attack, attackers can obtain a sensor's password of any length and combination. Additionally, by enumerating all possible MACaddresses, the attacker can launch a device scanning attack to find all online sensors.
- *Sybil attack:* here, a single malicious node asserts multiple identities (so-called Sybil nodes) and is located at different places in the network. This leads to a colossal, unfair resource allocation [56].

Roles—User Identification

- *Password attack:* there are a number of methods that can be used to gain unauthorized access to a password [57].

4. Attacks and Countermeasures

This section applies the attacks identified in Section 4.1 to the proposed blockchain architecture and discusses the resulting impact on the infrastructure. The countermeasures described in Section 4.2 are listed. These can be applied in securing the various systems or blockchain technologies. Finally, an overview in the form of a juxtaposition matrix is briefly presented in Section 4.3.

4.1. Attacks

4.1.1. Infrastructure- Hardware

The blockchain is connected to systems to interact with production machines and user devices. The hardware on which this infrastructure runs can be attacked.

Node Tampering

- *Condition monitoring:* the monitoring system is physically accessed and manipulated so that it stops monitoring or monitors the wrong machine, giving a false sense of security. The opposite is also possible (causing false alarms).
- *Legacy machine:* the machine connected to the network is physically accessed and tampered with. The hardware is modified by removing security features. Malicious control systems that affect the performance or reliability of the legacy equipment are inserted.
- *Network gateway:* the gateway for the attacker's next steps is physically accessed and manipulated. Malicious circuits that affect the performance or reliability of an electronic system are inserted.
- *OPC UA server:* the OPC UA server is physically accessed and manipulated to connect additional/incorrect input devices (e.g., simulate simple sensors or complete machines).
- *Sensor:* the door detection sensor is removed so that the door remains open while the machine is in operation. A new sensor is physically added to the network. For example, a malicious sensor could be added by replicating the identification of a sensor. This could lead to performance degradation or allow the attacker to inject spoofed data into the sensor network.

Malicious Code Injection

- *Legacy machine, OPC UA server, and sensor:* an attacker could gain access to any of these components by installing a malicious industrial add-in. When using debug functions (such as Joint Test Action Group (JTAG) pins that are still active), it is even possible to modify executed code and cause the device to malfunction.
  These three targets are explained together because of their similarities.

Sleep Denial Attack

- *Sensor:* attackers keep the battery-powered devices awake by feeding them with false inputs. This leads to the exhaustion of their batteries and thus to their shutdown.

4.1.2. Infrastructure—Software

Because a variety of software are used to operate the blockchain and related systems, attacks on that software must be considered.

Malware: Adware, Spyware, Trojan, Virus and Worms

- *Auxiliary DB:* by infecting the Auxiliary DB, the attacker can delete or encrypt data. Data can be manipulated without leaving any external or internal traces (no logs, etc. except the malware itself). By being already inside the trusted network, either data can be sent to external servers controlled by the attacker or other internal targets can be attacked.
- *Condition monitoring:* by having control of the system, an attacker can either disable alarms or create fake alarms, either completely destroying already defective machines or completely shutting down fully functional machines. The attacker can also replace the trained model or gain access to confidential model data.
- *OPC UA server:* a malware infection of the OPC UA server enables the attacker to send forged data to the connected systems (e.g., the blockchain). On the other hand, they are also able to control the machine to the point of damaging it by sending it fake or implausible commands.
- *Sensor:* by infecting the sensor arrays, it is possible to provide fake data that cannot be detected using the other systems. The falsified data can be used to shut down machines or to manipulate them directly into the production process.

Timejacking

- *Client to peers:* if a client has an incorrect time, the specified transaction period is no longer valid and the transaction is immediately rejected.
- *Peers in blockchain:* if a peer has a different time because it uses a compromised NTP server, the result is similar: the proposed transaction will fail with that particular peer because the valid time frame can no longer be guaranteed. Therefore, depending on the transaction policy, the whole transaction may be rejected.

4.1.3. Communication Channel

In addition to the systems themselves, the communication between them can also be attacked directly.

Routing

- *Condition monitoring and legacy machine:* messages are redirected to generate traffic at the network gateway and to keep the network busy. The redirected traffic can also be intercepted and stored by an attacker. *These two targets are explained together because of their similarities.*
- *Network gateway:* attacks in which an attacker forges or modifies routing information causes annoyances such as delivery error messages. Sending a large amount of messages to the blockchain exhausts all resources by responding to the spoofed traffic, making them unable to process legitimate service requests.
- *OPC UA server and sensor:* data messages are redirected to either access them from outside or to try to overload the network with them. *These two targets are explained together because of their similarities.*

Replay

- *Client to peers:* when sending a captured transaction twice, it is theoretically possible to execute the same chaincode twice and thus to change the state of the ledger (double spending problem). Fabric requires users to send a generated nonce value that has been deterministically hashed with the previous nonce value.
- *Peers in blockchain:* since the validators track the nonce value for each registered user, a repeated transaction can be detected and will be rejected on the network.
- *Sensor:* an attacker could record a signed packet and send the packet to the target multiple times to manipulate the system.

Selective Forwarding

- *Client to peers:* an attacker can selectively discard transactions to the nodes. Therefore, the information that reaches the destination is incomplete. Later, when another transaction reaches the node, Fabric detects that transactions are missing.

4.1.4. Data

The content of the data can be changed in various ways.

Account Hijacking

- *Condition monitoring and OPC UA server:* an attacker uses compromised/stolen credentials to gain access and to impersonate the account. Typically, account hijacking is done through phishing, sending fake emails to the user, password guessing, or a variety of other tactics. By accessing these systems, it is possible to easily send valid fake data. An attacker could also compromise the active session and gain access to the transmitted data.
  These two targets are explained together because of their similarities.

Brute Force Attack

- *Condition monitoring and OPC UA server:* the attacker tries to gain access to simple password-protected data by randomly trying passwords. Since these systems have no rate limiting, the attacker will eventually succeed. *These two targets are explained together because of their similarities.*

Data Inconsistency

- *Auxiliary DB:* read and write access enables various actions such as reading, modifying, or deleting the information. Data inconsistency can also occur if the VPN tunnel crashes and the data in the blockchain and the Auxiliary DB are no longer synchronized.

Malicious Smart Contracts

- *Peers in blockchain:* Since a smart contract is simply a piece of code that is executed on the peers, it is possible to exploit flaws in the code by transmitting specially crafted data. Furthermore, if a malicious smart contract is installed on the network, it can manipulate the data it receives.

Sensor Data/User Data

- *Network gateway:* attackers can steal important information, including passwords and intellectual property, by gaining access to the network. It is possible to steal user identities, to send the wrong information to the system, or to manipulate the data by impersonating an authorized user.
- *OPC UA server:* by modifying OPC UA data, it is possible to trigger data checks that can either shut down the machine or at least falsify information about the product, leading to quality problems.
- *Sensor:* radio signals can interfere with communication between the sensors and the device by affecting the signal-to-inference ratio, resulting in intentional (or unintentional) crosstalk.

Unauthorized Access

- *Auxiliary DB:* unauthorized access by a malicious actor can result in altered data. The attacker can also retrieve confidential information (e.g., bulk sensor/process data from machines).

4.1.5. Encryption

Encryption methods are used to protect the confidentiality of information. Unauthorized changes can be prevented and detected. Nevertheless, there may be vulnerabilities, either in the algorithm or, more commonly, in a particular implementation.

Side Channel Attack

- *OPC UA server:* the attacker gains access to the encryption keys and certificates used to secure the OPC UA connections and can thus read encrypted data.
- *Peers in blockchain:* the attacker collects the communication encryption keys by applying timing, power, error attacks, etc. to the devices running the blockchain nodes.

4.1.6. Roles—Device Identification

Spoofing

- *Sensor:* to fake sensor values, the identity of the sensor can be faked so that the device providing the data is not the sensor but, for example, a small computer that generates plausible values.

Sybil Attack

- *Auxiliary DB:* multiple fake identities can be used to easily overwhelm DB access control lists when it is easy to create new valid users due to the lack of strong authentication methods (unlike blockchain access control).

4.1.7. Roles—Password Attack

- *Auxiliary DB, condition monitoring, and OPC UA server:* password recovery attacks are performed to reset passwords to gain unauthorized access to the system [57].
  These three targets are explained together because of their similarities.

4.2. Countermeasures

This section provides an overview of countermeasures that can be applied to mitigate or completely prevent a successful attack on the BISS:4.0 platform. The countermeasures described are deliberately brief because they apply to many attacks.

- *C-Backups* ensure regular backups and provide security measures to prevent anonymous persons from entering the workshop area.
- *C-Audit:* while the most mentioned countermeasures are designed to preemptively protect against attacks, the countermeasure of an audit is used to evaluate the security of a given infrastructure, system, or software at a given snapshot in time. It indentifies exposed existing security vulnerabilities to prevent future attacks. There are a wide variety of security audit standards that exist; the most common utilized are ISO/IEC 27001 [58], NIST Cybersecurity Framework [59], Cyber Essentials [60], and BSI IT-Grundschutz [61].
- *C-Smart-Contract:* smart contracts are a very impactful source of attacks. Research has developed a wide variety of countermeasures for a wide variety of blockchain platforms. The most common approach is related to static code analysis that checks for patterns that are known causes [62,63]. These tools are mostly in early developmental states and are not available for the general public.
- *C-Optimization:* fuzzy logic and ant colony optimization for jamming attack detection are proposed.
- *C-Tracking:* a tracking mechanism, such as a sequence number, is used to identify packets that have already been processed but retransmitted with potentially malicious commands and data.
- *C-Identity-Verification:* identities must be verified. This can be done by encrypting with public keys or by granting access with the identity registered in the Blockchain.
- *C-Restrict-Access:* access can be restricted at two levels: restricted access to the building in which the infrastructure is located and restricted access at the account level of computer systems and the software. The goal is to prevent unauthorized individuals from accessing a resource. There are standards that specify approved methods for establishing restricted access to a building. In Germany, the approved methods are part of the BSI IT-Grundschutz [61]. Methods to protect against unauthorized access to computer systems and software can include password, two-factor authentication, smart cards, or biometrics. In Germany, the approved methods are part of BSI IT-Grundschutz [61].
- *C-Check-Connection:* once a connection has been established, whether the connection is still functional is checked, which can be performed with a heartbeat mechanism.
- *C-Monitor-Traffic* monitors inbound and outbound traffic for anomalies, which can be performed by either only looking at the metadata (e.g., IP header) or by looking at the content (deep packet inspection).
- *C-Encryption:* it can be prevented by encrypting and authenticating the superficial link layer using a standard global key.
- *C-Isolation:* as discussed in Section 3.1, smartphones can be infected and made accessible to attackers via installable applications. Isolation point mechanisms must be implemented for software running in development environments. Mobile applica-

tions, for example, must notify and request permission before accessing storage or network resources. The logic of automation and active software components should be the same because they are not reliable and should never be considered reliable elements.

- *C-Concealment:* obfuscation disguises the intended meaning of the communication by making the message difficult to understand.
- *C-Vulnerability-Scan:* a preventive action is the scan for vulnerabilities. One method is fuzzing software; this includes generating random input for the target system, revealing failures due to stress. The network connected systems can be scanned for open ports. This data is a foundation for securing the systems.
- *C-Security-Analysis:* vulnerabilities at the system boundaries are searched for. The used software/libraries are checked for known CVEs (CVE monitoring), and the software is updated accordingly to mitigate them. Additionally, new software/library releases are checked regardless (can be integrated with a CI run to automate the monitoring/checking).
- *C-Attack-Defence:* data, programming language, sandbox, virtual machine (VM) and operating system (OS), machine learning (ML) on physical data, coverage of k-Nearest Neighbors (kNN) algorithm, random forest algorithm, and anomaly detection algorithm are isolated to detect malicious attacks in real time.
- *C-Physical-Protection:* physical protection includes using shielded cables for physical connections and use of separate racks or rooms.
- *C-Secure-Connection:* a connection that employs encryption and authentication (e.g., VPN and TLS) is used.
- *C-Antivirus:* antivirus and antimalware vulnerability are assessed (the process of defining, identifying, classifying, and prioritizing vulnerabilities in computer systems, applications, and network infrastructures).
- *C-Code-Analysis:* static source code analysis can be used to find coding errors, which can potentially lead to vulnerabilities. Furthermore, automatic testing of the codebase with dynamic code testing tools can uncover runtime issues.
- *C-Penetration-Testing:* penetration testing is used to evaluate and exploit vulnerabilities.

4.3. Juxtaposition of Attacks and Countermeasures

This section provides a table showing which attack can be addressed with which countermeasure. After identifying attacks against the use case and developing countermeasures against those attacks, a juxtaposition of the attacks and applicable countermeasures is presented in Table 2, and the appropriate countermeasures are marked with an X in the table; see Table 1 for reference to the indexed attacks, and see Section 4.2 for reference to the countermeasures against the attacks. As an example, the index A-IH-NT-CM represents an attack on a condition monitoring system that is assigned to the attack type infrastructures—hardware—node tampering. The described attack can be weakened or even prevented by the countermeasures C-Attack-Defence, C-Audit, C-Penetration-Testing, C-Physical-Protection, C-Restrict-Access, C-Security-Analysis, and C-Vulnerability-Scan.

Table 2. Juxtaposition matrix of attacks and countermeasures.

Attacks \ Countermeasures	C-Antivirus	C-Attack-Defence	C-Audit	C-Backups	C-Concealment	C-Check-Connection	C-Encryption	C-Identity-Verification	C-Isolation	C-Optimization	C-Monitor-Traffic	C-Penetration-Testing	C-Physical-Protection	C-Restrict-Access	C-Secure-Connection	C-Security-Analysis	C-Smart-Contract	C-Code-Analysis	C-Tracking	C-Vulnerability-Scan
A-IH-NT-CM		X	X								X	X	X		X				X	
A-IH-NT-LM		X	X								X	X	X		X				X	
A-IH-NT-NG		X	X								X	X	X		X				X	
A-IH-NT-OPC		X	X								X	X	X		X				X	
A-IH-NT-SN		X	X								X	X	X		X				X	
A-IH-MCI-LM	X	X	X					X			X				X		X		X	
A-IH-MCI-OPC	X	X	X					X			X				X		X		X	
A-IH-MCI-SN	X	X	X					X			X				X		X		X	
A-IH-SDA-LM		X	X		X		X			X	X				X		X		X	
A-IS-MW-ADB	X	X	X					X		X	X				X					
A-IS-MW-CM	X	X	X					X		X	X				X				X	
A-IS-MW-OPC	X	X	X					X		X	X				X				X	
A-IS-MW-SN	X	X	X					X		X	X				X				X	
A-IS-TJ-CP			X		X		X							X	X				X	
A-IS-TJ-PB			X		X		X							X	X				X	
A-CC-RT-CM		X			X		X			X	X			X	X			X	X	
A-CC-RT-LM		X			X		X			X	X			X	X			X	X	
A-CC-RT-NG		X			X		X			X	X			X	X			X	X	
A-CC-RT-OPC		X			X		X			X	X			X	X			X	X	
A-CC-RT-SN		X			X		X			X	X			X	X			X	X	
A-CC-RP-CP		X			X		X			X				X				X	X	
A-CC-RP-PB		X			X		X			X				X				X	X	
A-CC-RP-SN		X			X		X			X				X				X	X	
A-CC-SF-CP		X			X					X				X	X			X	X	
A-D-AH-CM		X	X				X			X	X				X		X	X	X	
A-D-AH-OPC		X	X				X			X	X				X		X	X	X	
A-D-BFA-CM		X	X		X			X	X	X	X				X		X		X	
A-D-BFA-OPC		X	X		X			X	X	X	X				X		X		X	
A-D-DI-ADB			X	X				X					X	X	X		X		X	
A-D-MSC-PB			X				X	X			X				X	X	X		X	
A-D-SD-NG			X		X		X	X			X		X	X	X					
A-D-SD-OPC			X		X		X	X			X		X	X	X					
A-D-SD-SN			X		X		X	X			X		X	X	X					
A-D-UA-ADB		X	X				X				X		X	X	X				X	
A-E-SCA-OPC		X	X		X	X	X			X	X			X	X			X	X	
A-E-SCA-PB		X	X		X	X	X			X	X			X	X			X	X	
A-RDI-SF-SN		X	X		X	X	X			X	X			X	X				X	
A-RDI-SA-ADB		X	X		X		X				X			X	X				X	
A-RUI-SA-ADB		X	X				X				X							X		
A-RUI-PA-ADB		X	X			X	X				X				X		X	X	X	
A-RUI-PA-CM		X	X			X	X				X				X		X	X	X	
A-RUI-PA-OPC		X	X			X	X				X				X		X	X	X	

5. Discussion: Application of BISS:4.0 to Real-World Scenarios

This section describes a discussion of four real-world scenarios that can be executed safely, even while being under attack, by using the BISS:4.0 platform. Attack vectors are combined into an attack path; their effects and how to prevent bad outcomes of these attacks are described. In scenarios where complete protection against attacks is not possible, the limits of the solution's effectiveness are described.

5.1. Scenario 1: Sabotage of the Production Line

- *Intention:* in this attack scenario, a malicious insider attempts to sabotage a production line via infrastructure hardware node tampering of an OPC UA server that connects sensors to a condition monitoring system.
- *Attack path and impact:* the attacker uses a lighter to suddenly increase the value of a temperature sensor in a production machine to manipulate the actions performed by the condition monitoring system to initiate an unnecessary shutdown of the production line. The condition monitoring system would respond to this temperature rise by shutting down the production machine with which the sensor has been tampered.
- *Countermeasure:* as a countermeasure, the control systems of production machines are connected to the Hyperledger Fabric blockchain, which is the core component of the BISS:4.0 platform. Hyperledger Fabric logs the sensor data for the purpose of collecting evidence and executing smart contracts. A smart contract performs temperature plausibility checks to determine if the reported sensor value is even close to being realistic. If there is an unrealistic temperature rise reported by the temperature sensor, the production machine is not shut down immediately, and the blockchain issues an event that causes a service technician to check the temperature sensor, since the unrealistic temperature rise could also be a faulty sensor. If there is a realistic temperature rise reported by the temperature sensor, the production machine is shut down and the blockchain issues an event that causes a service technician to check the machine.

5.2. Scenario 2: Multi-Worm Attack

- *Intention:* in this attack scenario, an attacker uses a multi-worm attack to destroy production equipment.
- *Attack path and impact:* the infection occurs via a portable data storage device and spreads through exploitations in the operating system. The virus searches the infected host for software that can access programmable logic controllers (PLCs). PLCs are used for automation and monitoring of electromechanical devices. The malware updates itself, especially the specific attack code. The goal of the attack code is to damage the electromechanical equipment controlled by the host. At the same time, the virus sends false feedback to the main controller. Therefore, the monitoring systems cannot detect the misuse until the self-destruction of the equipment has started [45].
- *Countermeasure:* BISS:4.0 with its blockchain technology can prevent attacks by securing two attack surfaces. The first is the misuse of devices. With blockchain technology, the programs to execute processes of the devices are stored in the blockchain and are thus protected from tampering. The second prevented attack surface is the false feedback. Sensor data are written directly from the sensors or sensor systems to the blockchain. Another countermeasure such as encapsulating the systems connected to the devices, such as using read-only systems to execute device code and read-only sensor systems or banning all portable media and using security software to intercept malware before it can be transmitted over the network, could also be a feasible approach but often cannot be consistently enforced.

5.3. Scenario 3: Careless Maintenance Technician

- *Intention:* in this attack scenario, a careless maintenance technician does not correctly execute the maintenance process of machines. Although this attack does not fit into the

identified attack categories, this attack demonstrates the scope of protection offered by BISS:4.0.

- *Attack path and impact:* the maintenance technician performs maintenance tasks and logs them (tasks performed, material consumed, time required to perform the task, etc.) for the purpose of documentation and accounting. As an example, the maintenance task requires the maintenance technician to wear safety glasses and gloves to comply with health and safety regulations. In addition, the maintenance task specifies that a certain type of lubricating oil must be used. The careless maintenance technician fails to wear the required safety equipment, resulting in liability in the event of an accident that could have been prevented by complying with workplace safety regulations. In addition, the careless maintenance technician performs the maintenance task incorrectly by using the wrong type of lubricating oil, which may result in liability in the event of a machine failure.
- *Countermeasure:* the BISS:4.0 platform defines a countermeasure against negligence of wearing the safety equipment by utilizing a safety equipment storage box that is equipped with sensors connected to the blockchain to log the opening and closing of the storage box. While this does not guarantee that the maintenance technician actually wears the security equipment, it increases the likelihood that he or she will do so, as the immutable record of the operation's log on the blockchain obligates him or her to comply. The second countermeasure that the BISS:4.0 platform defines is against the threat of using the wrong type of lubricating oil: the oil canister is equipped with an RFID tag that contains information about the oil and is read by a tablet or smartphone before the oil is put into the machine. The reading of the RFID tag by the tablet or smartphone triggers a check within the blockchain via a smart contract to ensure that the correct lubricating oil grade is used. In the event that the wrong type of oil is used, the maintenance technician is notified of their incorrect choice. While this does not guarantee that the maintenance technician will actually use the correct type of lubricating oil, it, again, increases the likelihood that he or she will do so by committing the technician to perform the maintenance task correctly by immutably recording the log of the operation in the blockchain.

5.4. Scenario 4: Unwitting Disclosure of Confidential Data

- *Intention:* this attack scenario describes an attack that has already occurred in the real world with widespread impact [64] and could have been averted or mitigated through the use of the BISS:4.0 platform. Although this attack does not fit into the identified attack categories, this attack demonstrates the scope of protection offered by BISS:4.0. Although this was not a targeted attack, "just a case of carelessness", it clearly demonstrates how easily a malicious insider or external attacker can grab a company's secret data through an official data channel without them even realizing it.
- *Attack path and impact:* in this real-world scenario, a robotics manufacturer unknowingly disclosed confidential data (assembly line layouts, employee data, non-disclosure agreements, etc.) from a number of large manufacturing companies through its data exchange infrastructure.
- *Countermeasure:* in an earlier paper [65], a blockchain-based approach that creates and manages an audit trail of data exchange through the blockchain without requiring the data to be sent through the blockchain for the privacy validation was described. Instead, the confidentiality of the data was validated by a privacy validation module and the validated data was then exchanged between the companies involved in the data exchange via P2P communications. To ensure the trust and effectiveness of the modules, they were transparently validated and certified to users before use. In addition, a method for processing confidential data with smart contracts that trigger specific actions based on inputs without disclosing the data was described. Hash-based mapping was used to assign the confidential data to hash values, which were then compared against predefined trigger conditions that then triggerred a specific

action (e.g., triggering a maintenance task). This approach, which validates outbound data against a privacy policy before it is sent, prevents, or at least makes it more difficult for a malicious insider to leak the company's secret data through an official data channel.

6. Conclusions

In this paper, an overview was created for companies that want to integrate blockchain technology into their existing production environment with legacy machines and retrofit IoT-like sensors for data collection. These newly created communication channels, which did not exist before, create new and, presumably, previously existing but unknown attack opportunities. Information about these attack possibilities is provided vividly by means of the described BISS:4.0 platform, which pursues the goal of blockchain integration in existing environments. A security audit was conducted to identify the attack possibilities, to classify them, and to describe them using examples for each type of attack. Identified attack possibilities are contrasted with a number of already successfully used countermeasures as well as countermeasures emerging from blockchain technology, such as the use of distributed smart contracts for the purpose of sensor data validation. Finally, four real-world scenarios were described that illustrate an attack, its intent, its execution and impact, and countermeasures against the attack, showcasing the real-world security benefit of integrating blockchain into an industrial environment.

Author Contributions: Conceptualization, A.S. and C.R.; funding acquisition, A.S. and C.R.; investigation, J.S., D.S., D.W. and F.G.G.; methodology, J.S., D.S., D.W. and F.G.G.; project administration, A.S. and C.R.; visualization, F.G.G. and D.S.; writing—original draft J.S., D.S. and D.W.; writing—review and editing, A.S. and C.R. All authors have read and agreed to the published version of the manuscript.

Funding: This work has received funding from European Fonds for Regional Development (EFRE) and the Ministry of Science, Research, and Art of Baden-Württemberg (MWK) in the framework of the project BISS:4.0 (biss40.in.hs-furtwangen.de).

Data Availability Statement: Not Applicable, the study does not report any data.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AEAD	Authenticated Encryption with Associated Data
CLI	Command Line Interface
DDoS	Distributed Denial of Service
DNS	Domain Name System
I2C	Inter-Integrated Circuit
IoT	Internet of Things
JTAG	Joint Test Action Group
OCI	Open Container Initiative
OEM	Original Equipment Manufacturer
OPC UA	Open Platform Communications Unified Architecture
PDoS	Permanent Denial of Service
PLC	Programmable Logic Controller
P2P	Peer-to-Peer
RFID	Radio-Frequency Identification
SDK	Software Development Kit
SSL	Secure Sockets Layer
TLS	Transport Layer Security
VPN	Virtual Private Network

References

1. Herterich, M.M.; Buehnen, T.; Uebernickel, F.; Brenner, W. A Taxonomy of Industrial Service Systems Enabled by Digital Product Innovation. In Proceedings of the 2016 49th Hawaii International Conference on System Sciences (HICSS), Koloa, HI, USA, 5–8 January 2016; pp. 1236–1245. [CrossRef]
2. Macaulay, T.; Singer, B.L. *Cybersecurity for Industrial Control Systems: SCADA, DCS, PLC, HMI, and SIS*; CRC Press: Boca Raton, FL, USA, 2011.
3. Marhaug, A.; Schjølberg, P. Smart Maintenance-Industry 4.0 and Smart Maintenance: From Manufacturing to Subsea Production Systems. In Proceedings of the 6th International Workshop of Advanced Manufacturing and Automation, Manchester, UK, 10–11 November 2016; Atlantis Press: Paris, France, 2016.
4. Thoben, K.D.; Wiesner, S.; Wuest, T. "Industrie 4.0" and smart manufacturing-a review of research issues and application examples. *Int. J. Autom. Technol.* **2017**, *11*, 4–16. [CrossRef]
5. Waidner, M.; Kasper, M. Security in industrie 4.0-challenges and solutions for the fourth industrial revolution. In Proceedings of the 2016 Design, Automation & Test in Europe Conference & Exhibition (DATE), Dresden, Germany, 14–18 March 2016; pp. 1303–1308.
6. Ervural, B.C.; Ervural, B. Overview of cyber security in the industry 4.0 era. In *Industry 4.0: Managing the Digital Transformation*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 267–284.
7. Pan, F.; Pang, Z.; Luvisotto, M.; Xiao, M.; Wen, H. Physical-layer security for industrial wireless control systems: Basics and future directions. *IEEE Ind. Electron. Mag.* **2018**, *12*, 18–27. [CrossRef]
8. Tuptuk, N.; Hailes, S. Security of smart manufacturing systems. *J. Manuf. Syst.* **2018**, *47*, 93–106. [CrossRef]
9. Andrea, I.; Chrysostomou, C.; Hadjichristofi, G. Internet of Things: Security vulnerabilities and challenges. In Proceedings of the 2015 IEEE Symposium on Computers and Communication (ISCC), Larnaca, Cyprus, 6–9 July 2015; pp. 180–187.
10. Ahemd, M.M.; Shah, M.A.; Wahid, A. IoT security: A layered approach for attacks & defenses. In Proceedings of the 2017 international conference on Communication Technologies (ComTech), Rawalpindi, Pakistan, 19–21 April 2017; pp. 104–110.
11. Sengupta, J.; Ruj, S.; Bit, S.D. A Comprehensive survey on attacks, security issues and blockchain solutions for IoT and IIoT. *J. Netw. Comput. Appl.* **2020**, *149*, 102481. [CrossRef]
12. Baskar, R.; Raja, P.; Reji, M.; Joseph, C. Performance analysis of scalability in the sinkhole compromised topology of wireless sensor networks. *Int. J. Pure Appl. Math.* **2017**, *117*, 35–39.
13. Varga, P.; Plosz, S.; Soos, G.; Hegedus, C. Security threats and issues in automation IoT. In Proceedings of the 2017 IEEE 13th International Workshop on Factory Communication Systems (WFCS), Trondheim, Norway, 31 May–2 June 2017; pp. 1–6.
14. Januário, F.; Carvalho, C.; Cardoso, A.; Gil, P. Security challenges in SCADA systems over Wireless Sensor and Actuator Networks. In Proceedings of the 2016 8th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), Lisbon, Portugal, 18–20 October 2016; pp. 363–368.
15. Machado, C.; Fröhlich, A.A.M. IoT data integrity verification for cyber-physical systems using blockchain. In Proceedings of the 2018 IEEE 21st International Symposium on Real-Time Distributed Computing (ISORC), Singapore, 29–31 May 2018; pp. 83–90.
16. Maynard, P.; McLaughlin, K.; Sezer, S. Using Application Layer Metrics to Detect Advanced SCADA Attacks. In Proceedings of the ICISSP, Funchal, Portugal, 22–24 January 2018; pp. 418–425.
17. Tanwar, S.; Kumar, A. A proposed scheme for remedy of man-in-the-middle attack on certificate authority. *Int. J. Inf. Secur. Priv. (IJISP)* **2017**, *11*, 1–14. [CrossRef]
18. Schönle, D.; Wallis, K.; Stodt, J.; Reich, C.; Welte, D.; Sikora, A. Industry Use Cases on Blockchain Technology. In *Industry Use Cases on Blockchain Technology Applications in IoT and the Financial Sector*; Mahmood, Z., Ed.; IGI Global: Hershey, PA, USA, 2021. [CrossRef]
19. Androulaki, E.; Barger, A.; Bortnikov, V.; Cachin, C.; Christidis, K.; Caro, A.D.; Enyeart, D.; Ferris, C.; Laventman, G.; Manevich, Y.; et al. Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains. *arXiv* **2018**, arXiv:1801.10228.
20. Maggi, F.; Pogliani, M. Attacks on Smart Manufacturing Systems. Available online: https://resources.mynewsdesk.com/image/upload/t_attachment/dc5ixveews0zqu6txtni.pdf (accessed on 7 April 2021).
21. Homoliak, I.; Venugopalan, S.; Reijsbergen, D.; Hum, Q.; Schumi, R.; Szalachowski, P. The Security Reference Architecture for Blockchains: Towards a Standardized Model for Studying Vulnerabilities, Threats, and Defenses. *IEEE Commun. Surv. Tutor.* **2020**. [CrossRef]
22. Wisseman. Third-Party Libraries Are One of the Highest Security Risks. Available online: https://techbeacon.com/security/third-party-libraries-are-one-most-insecure-parts-application (accessed on 15 February 2019).
23. Bongiorni, L. A Remotely Controlled Malicious Usb Hid Injecting Cable. Available online: https://medium.com/@LucaBongiorni/usbsamurai-a-remotely-controlled-malicious-usb-hid-injecting-cable-for-less-than-10-ebf4b81e1d0b (accessed on 15 December 2012).
24. Tang, C.; Chen, S.; Fan, L.; Xu, L.; Liu, Y.; Tang, Z.; Dou, L. A large-scale empirical study on industrial fake apps. In Proceedings of the 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP), Montreal, QC, Canada, 25–31 May 2019; pp. 183–192.
25. Tedeschi, S.; Emmanouilidis, C.; Farnsworth, M.; Mehnen, J.; Roy, R. New threats for old manufacturing problems: Secure IoT-Enabled monitoring of legacy production machinery. In Proceedings of the IFIP International Conference on Advances in Production Management Systems, Austin, TX, USA, 1–5 September 2019; Springer: Berlin/Heidelberg, Germany, 2017; pp. 391–398.

26. Kovacs, E. Many Vulnerabilities Found in OPC UA Industrial Protocol. Available online: https://www.securityweek.com/many-vulnerabilities-found-opc-ua-industrial-protocol (accessed on 15 March 2020).
27. Nedeljković, D.; Jakovljević, Ž.; Miljković, Z. The detection of sensor signal attacks in industrial control systems. *FME Trans.* **2020**, *48*, 7–12. [CrossRef]
28. Trustwave. 2018 Trustwave Global Security Report. Available online: https://www.trustwave.com/en-us/resources/library/documents/2018-trustwave-global-security-report/ (accessed on 15 April 2018).
29. Specht, F.; Otto, J.; Niggemann, O.; Hammer, B. Generation of adversarial examples to prevent misclassification of deep neural network based condition monitoring systems for cyber-physical production systems. In Proceedings of the 2018 IEEE 16th International Conference on Industrial Informatics (INDIN), Porto, Portugal, 18–20 July 2018; pp. 760–765.
30. Poston, H. Attacks on Blockchain. Available online: https://resources.infosecinstitute.com/topic/attacks-on-blockchain/ (accessed on 15 September 2020).
31. Kovacs, E. Vulnerabilities in Protocol Gateways Can Facilitate Attacks on Industrial Systems. Available online: https://www.securityweek.com/vulnerabilities-protocol-gateways-can-facilitate-attacks-industrial-systems (accessed on 15 March 2020).
32. Messaad, M.A.; Jerad, C.; Sikora, A. AI Approaches for IoT Security Analysis. In Proceedings of the International Conference on Applied Soft Computing and Communication Networks (ACN'20), Chennai, India, 14–17 October 2020.
33. Möller, B.; Duong, T.; Kotowicz, K. This POODLE Bites: Exploiting the SSL 3.0 fallback. Available online: https://www.openssl.org/~bodo/ssl-poodle.pdf (accessed on 7 April 2021).
34. ImperialViolet. The POODLE Bites Again. Available online: https://www.imperialviolet.org/2014/12/08/poodleagain.html (accessed on 14 December 2020).
35. OpenSSL. CVE-2014-0160–Heartbleed. Available online: https://www.openssl.org/news/secadv/20140407.txt (accessed on 14 December 2020).
36. Alladi, T.; Chamola, V.; Sikdar, B.; Choo, K.K.R. Consumer IoT: Security vulnerability case studies and solutions. *IEEE Consum. Electron. Mag.* **2020**, *9*, 17–25. [CrossRef]
37. Jardine, W.; Frey, S.; Green, B.; Rashid, A. Senami: Selective non-invasive active monitoring for ics intrusion detection. In Proceedings of the 2nd ACM Workshop on Cyber-Physical Systems Security and Privacy, Vienna, Austria, 28 October 2016; pp. 23–34.
38. Yoo, H.; Ahmed, I. Control logic injection attacks on industrial control systems. In Proceedings of the IFIP International Conference on ICT Systems Security and Privacy Protection, Maribor, Slovenia, 21–23 September 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 33–48.
39. Gallais, A.; Hedli, T.H.; Loscri, V.; Mitton, N. Denial-of-Sleep Attacks against IoT Networks. In Proceedings of the 2019 6th International Conference on Control, Decision and Information Technologies (CoDIT), Paris, France, 23–26 April 2019; pp. 1025–1030.
40. Yamaguchi, S.; Gupta, B. Malware threat in Internet of Things and its mitigation analysis. In *Security, Privacy, and Forensics Issues in Big Data*; IGI Global: Hershey, PA, USA, 2020; pp. 363–379.
41. Khajanchee, T.; Kshirsagar, D. Attacks on Blockchain-Based Systems. In *Blockchain Technology and the Internet of Things: Challenges and Applications in Bitcoin and Security*; Apple Academic Press: Palm Bay, FL, USA, 2020; p. 201.
42. Jamai, I.; Azzouz, L.B.; Saïdane, L.A. Security issues in Industry 4.0. In Proceedings of the 2020 International Wireless Communications and Mobile Computing (IWCMC), Limassol, Cyprus, 15–19 June 2020; pp. 481–488.
43. Alcaraz, C.; Bernieri, G.; Pascucci, F.; Lopez, J.; Setola, R. Covert channels-based stealth attacks in industry 4.0. *IEEE Syst. J.* **2019**, *13*, 3980–3988. [CrossRef]
44. Qian, J.; Du, X.; Chen, B.; Qu, B.; Zeng, K.; Liu, J. Cyber-physical integrated intrusion detection scheme in SCADA system of process manufacturing industry. *IEEE Access* **2020**, *8*, 147471–147481. [CrossRef]
45. Langner, R. Stuxnet: Dissecting a Cyberwarfare Weapon. *IEEE Secur. Priv.* **2011**, *9*, 49–51. [CrossRef]
46. UmaRani, V.; Somasundaram, K. Detection of selective forwarding attack using BDRM in wireless sensor network. In *AIP Conference Proceedings*; AIP Publishing LLC: College Park, MD, USA, 2020; Volume 2271, p. 030029.
47. Ghasemisharif, M.; Ramesh, A.; Checkoway, S.; Kanich, C.; Polakis, J. O single sign-off, where art thou? an empirical analysis of single sign-on account hijacking and session management on the web. In Proceedings of the 27th {USENIX} Security Symposium ({USENIX} Security 18), Baltimore, MD, USA, 15–17 August 2018; pp. 1475–1492.
48. Haber, M.J. Attack Vectors. In *Privileged Attack Vectors*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 65–85.
49. Stiawan, D.; Idris, M.; Malik, R.F.; Nurmaini, S.; Alsharif, N.; Budiarto, R. Investigating Brute Force Attack Patterns in IoT Network. *J. Electr. Comput. Eng.* **2019**, *2019*, 4568368. [CrossRef]
50. Alem, S.; Espes, D.; Martin, E.; Nana, L.; De Lamotte, F. A hybrid intrusion detection system in industry 4.0 based on ISA95 standard. In Proceedings of the 2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA), Abu Dhabi, United Arab Emirates, 3–7 November 2019; pp. 1–8.
51. Dai, W.; Dai, C.; Choo, K.K.R.; Cui, C.; Zou, D.; Jin, H. SDTE: A secure blockchain-based data trading ecosystem. *IEEE Trans. Inf. Forensics Secur.* **2019**, *15*, 725–737. [CrossRef]
52. Liu, J.; Zhang, W.; Ma, T.; Tang, Z.; Xie, Y.; Gui, W.; Niyoyita, J.P. Toward security monitoring of industrial cyber-physical systems via hierarchically distributed intrusion detection. *Expert Syst. Appl.* **2020**, *158*, 113578. [CrossRef]

53. Lezzi, M.; Lazoi, M.; Corallo, A. Cybersecurity for Industry 4.0 in the current literature: A reference framework. *Comput. Ind.* **2018**, *103*, 97–110. [CrossRef]
54. Yu, S.Y.; Malawade, A.V.; Chhetri, S.R.; Al Faruque, M.A. Sabotage attack detection for additive manufacturing systems. *IEEE Access* **2020**, *8*, 27218–27231. [CrossRef]
55. Shoukry, Y.; Martin, P.; Yona, Y.; Diggavi, S.; Srivastava, M. Pycra: Physical challenge-response authentication for active sensors under spoofing attacks. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, 12–16 October 2015; pp. 1004–1015.
56. Wang, H. A Three-Tier Scheme for Sybil Attack Detection in Heterogeneous IWSN. Available online: https://www.researchgate.net/publication/339679791_A_three-tier_scheme_for_sybil_attack_detection_in_heterogeneous_IWSN (accessed on 7 April 2021).
57. Raza, M.; Iqbal, M.; Sharif, M.; Haider, W. A survey of password attacks and comparative analysis on methods for secure authentication. *World Appl. Sci. J.* **2012**, *19*, 439–444.
58. ISO—ISO/IEC 27001 — Information Security Management. Available online: https://www.iso.org/isoiec-27001-information-security.html (accessed on 14 February 2021).
59. NIST. Cybersecurity Framework. Available online: https://www.nist.gov/cyberframework (accessed on 14 February 2021).
60. Cyber Aware. Available online: https://www.ncsc.gov.uk/cyberaware/home (accessed on 14 February 2021).
61. IT-Grundschutz. Available online: https://www.bsi.bund.de/EN/Topics/ITGrundschutz/itgrundschutz.html;jsessionid=72AEED468D2781D1CCE9FE259EF54D3F.internet472?nn=409850 (accessed on 14 February 2021).
62. Huang, Y.; Bian, Y.; Li, R.; Zhao, J.L.; Shi, P. Smart Contract Security: A Software Lifecycle Perspective. *IEEE Access* **2019**, *7*, 150184–150202. [CrossRef]
63. Ghaleb, A.; Pattabiraman, K. How Effective Are Smart Contract Analysis Tools? Evaluating Smart Contract Static Analysis Tools Using Bug Injection. In Proceedings of the 29th ACM SIGSOFT International Symposium on Software Testing and Analysis, Los Angeles, CA, USA, 18–22 July 2020.
64. UpGuard Team. Short Circuit: How a Robotics Vendor Exposed Confidential Data for Major Manufacturing Companies. Available online: https://www.upguard.com/breaches/short-circuit-how-a-robotics-vendor-exposed-confidential-data-for-major-manufacturing-companies (accessed on 7 April 2021).
65. Stodt, J.; Reich, C. Data Confidentiality In P2P Communication And Smart Contracts Of Blockchain In Industry 4.0. *arXiv* **2020**, arXiv:2007.14195.