information

Article

Formal Security Analysis of ISA100.11a Standard Protocol Based on Colored Petri Net Tool

Tao Feng *, Taining Chen and Xiang Gong

School of Computer and Communication, Lanzhou University of Technology, Lanzhou 730050, China; chentn@lut.edu.cn (T.C.); gongxiang@lut.edu.cn (X.G.)
* Correspondence: fengt@lut.edu.cn

Abstract: This paper presents a formal security analysis of the ISA100.11a standard protocol using the Colored Petri Net (CPN) modeling approach. Firstly, we establish a security threat model for the ISA100.11a protocol and provide a detailed description and analysis of the identified security threats. Secondly, we use the CPN tool to model the protocol formally and conduct model checking and security analysis. Finally, we analyze and discuss the results of the model checking, which demonstrate that the ISA100.11a standard protocol may have vulnerabilities when certain security threats exist, and provide some suggestions to enhance the security of the protocol. This research provides a certain level of security assurance for the ISA100.11a standard protocol and serves as a reference for similar security research on protocols.

Keywords: CPN tools; ISA100.11a standard; colored Petri net theory; formal analysis

Citation: Feng, T.; Chen, T.; Gong, X. Formal Security Analysis of ISA100.11a Standard Protocol Based on Colored Petri Net Tool. *Information* **2024**, *15*, 118. https://doi.org/10.3390/info15020118

Academic Editor: Leandros Maglaras

Received: 2 January 2024
Revised: 24 January 2024
Accepted: 7 February 2024
Published: 18 February 2024

1. Introduction

With the continuous development and application of industrial automation technology, industrial control systems (ICSs) have become an essential component of modern industrial production. At the same time, the security issues of ICSs are receiving more and more attention. Since ICSs often operate in complex and open environments, once they are attacked or fail, they may have serious impacts on production processes, personnel safety, and environmental protection. Therefore, ensuring the security of ICSs has become an important task [1].

To improve the security of ICSs, various security strategies and technologies have been proposed. Formal methods have become an important technical means for the security analysis of ICSs. In the security analysis of ICSs, formal methods can be used to formally define the security attributes of the system, verify whether the security attributes of the system meet specific requirements, and analyze the security properties of the system.

The ISA100.11a standard is an important standard in the field of industrial automation, aimed at providing a secure, reliable, and efficient communication protocol for connecting and managing industrial equipment and sensors. To ensure the security of the ISA100.11a standard protocol, formal security analysis is required. The method based on Colored Petri Nets (CPNs) is a commonly used formal method, which has been widely applied in the modeling, analysis, and verification of ICSs.

ISA100.11a was developed by the ISA100 Wireless Compliance Institute under the International Society of Automation (ISA). The institute is dedicated to defining wireless system regulations and implementation technologies for industrial environments by establishing a series of standards, recommended operating procedures, and drafting technical reports [2]. The main contents of the ISA100.11a standard include the industrial wireless network architecture, coexistence, robustness, and interoperability with wired field networks. The industrial wireless devices defined by this standard include sensors, actuators, wireless handheld devices, and other field automation equipment. The ISA100.11a standard hopes to support industrial field applications with a low complexity, reasonable cost, low

power consumption, and appropriate communication data rates. Basic security services are provided between all devices by the ISA100.11a, and device internal security functions are mainly defined at the data link layer and transport layer. The security services include hash functions, symmetric ciphers, and asymmetric ciphers, and these security policies are provided by the security manager of the ISA100.11a network [3–6].

The security of ISA100.11a devices is based on internal encryption, which is considered more secure than data transmission between devices in the network. Therefore, this article aims to explore formal security analysis methods based on the Colored Petri Net (CPN) approach for the ISA100.11a standard protocol in a networked form to enhance its security and reliability.

This article is mainly divided into the following sections:

Section 2 provides an overview of the ISA100.11a standard protocol and the current research status of its related content. It explores the communication process of the ISA100.11a standard protocol and the application of Colored Petri Nets (CPNs) in the security analysis of Industrial Control Systems (ICS).

Section 3 introduces a method for the formal security analysis of the ISA100.11a standard protocol based on CPNs. It involves modeling and analysis, and through the examination and assessment of the model's state space report, it identifies security vulnerabilities in the protocol.

Section 4 proposes improvements to address the discovered vulnerabilities in the protocol. It models and analyzes the proposed enhancements and evaluates the security effectiveness of the new solutions through the analysis of the state space reports.

Section 5 outlines areas for further research in this study and provides prospects for future research on industrial wireless protocols.

In summary, this article aims to study a formal security analysis method for the ISA100.11a standard protocol based on CPNs in order to improve the security and reliability of the ISA100.11a standard protocol. Through the research in this article, a better understanding and application of formal methods can be achieved to analyze the security of ICSs, providing technical support and security guarantees for industrial control system security.

The main contributions of this paper include three aspects:

This security research of the ISA100.11a protocol adopts a formal model-checking approach based on Colored Petri Net theory and the Dolev-Yao attacker model.

This paper provides a detailed introduction to the ISA100.11a protocol, models the protocol using a CPN modeling tool, and verifies the consistency of the model. Based on the ISA100.11a standard, a mechanism for preventing security threats and attacks in ISA100.11a networks is studied. The Dolev-Yao attacker model is introduced to evaluate the security of the protocol and identify potential security vulnerabilities.

To address the security vulnerabilities in the protocol, a new and improved solution is proposed. This paper describes how to implement security services based on security measures and verifies the security of the new solution using an attack model.

2. Related Work

Ensuring the security of communication protocols is of the utmost importance in industrial control systems. Formal methods, such as a security analysis approach based on mathematical theory, have garnered significant attention in recent years. This paper employs a formal security analysis approach based on Colored Petri Nets (CPN) to analyze the security of the ISA100.11a standard protocol.

ISA100.11a represents a wireless communication protocol extensively employed in industrial control systems. Existing scholarly literature has conducted analyses on the security aspects of the ISA100.11a protocol. However, most studies have relied on simulation and testing methodologies, which possess certain limitations and lack formal assurance. For instance, in reference [7], researchers conducted an investigation into the communication process and encryption methods of this protocol, as illustrated in Figure 1. They

proposed a mechanism aimed at countering security threats and attacks in the ISA100.11a network, which standardized symmetric block ciphers and CCM* modes, and constructed an ISA100.11a-secure protocol stack, integrating wireless communication technologies. The packet security measures of this mechanism primarily verify the security processes of field devices and routers, encompassing the integrity of wireless data packets and secure transmissions. The researchers captured wireless data packets using the General Packet Sniffer software namede Chipcon General Packet Sniffer soft, performing an analysis and comparison of the payload to ultimately validate the integrity and correctness of the packets. Experimental results indicated that this mechanism exhibits commendable security and energy efficiency.

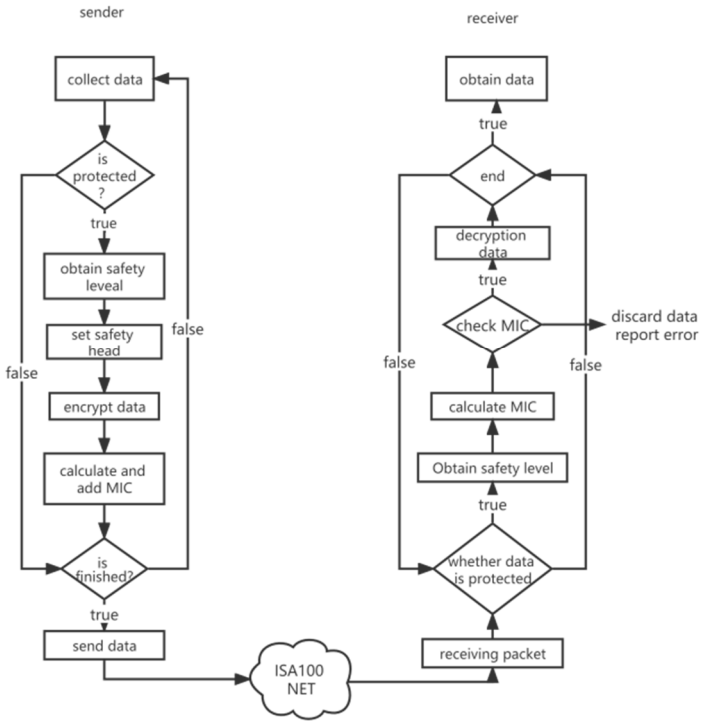

Figure 1. Protocol security for data processing.

Reference [8] evaluates ISA100.11a CSMA-CA using simulation, considering the effects of back-off procedures and priority settings to the probability of collision and successful use of slots. It is demonstrated that a high number of priority classes enable better network utilization resulting in a smaller number of packets exceeding their lifetime.

Simulation and testing methodologies primarily focus on system behavior within specific scenarios, unable to comprehensively explore all potential states and behaviors. In contrast, model checking based on the state space can offer a more comprehensive analysis.

In addition to simulation and testing-based approaches, some research work has also focused on the security of wireless communication protocols. For instance, researchers have introduced a three-factor user authentication protocol for wireless multimedia sensor networks (WMSN), as detailed in reference [9]. Their work claims that the proposed protocol not only ensures user anonymity but also prevents impersonation attacks by sensor nodes.

In the study outlined in reference [10], a lightweight anonymous identity verification and key agreement protocol for wireless body area networks (WBAN) was proposed, known as liteAuth. In their approach, they employ techniques such as random shuffling based on Tinkerbell mapping, Physical Unclonable Functions (PUFs), one-way hash functions, and bitwise exclusive OR operations to achieve mutual authentication and session key agreement.

However, despite the advancements made by these research works in enhancing the security of wireless communication protocols, there are still limitations and shortcomings. For instance, these methods may not cover all possible the states and attack paths of the protocol, and they may struggle to provide rigorous formal guarantees. Additionally, traditional security solutions may face challenges in running on lightweight devices due to their limitations in device chip memory and performance.

Furthermore, there have been explorations of the application of Colored Petri Nets (CPN) in formal modeling and security analysis methods for protocols. Due to the lack of a unified standard, there are various approaches utilizing CPN for protocol security analysis. In recent years, numerous works in the literature have presented different solutions. For instance, in reference [11], researchers employed a formal modeling approach, utilizing Colored Petri Nets (CPN) to construct an executable model of the DNP3 protocol. This model facilitates an in-depth analysis of the protocol's behavior, ensuring that it operates as expected. Through the formal modeling and analysis methods employed in this research, profound insights into the behavior of the DNP3 protocol were gained, particularly in the identification of potential security risks. This approach has proven to be significantly successful in shedding light on the intricacies of the DNP3 protocol, offering valuable contributions, especially in uncovering latent security threats.

Researchers in [12] proposed a compressive review of all the available methods for formal analysis along with CPN modeling that was completed for the analysis of valid as well as invalid states of the HART protocol. The modeling shows various states that the protocol can take during one transaction of communication. Further, this analysis can be extended for other Fieldbus protocols for security analysis.

Although previous research has made progress in the security of the ISA100.11a protocol, most methods still rely on simulation and testing, lacking formal guarantees. In contrast, model checking is a formal analysis method that involves establishing a protocol model and automatically verifying whether the model satisfies security properties such as authentication, confidentiality, and integrity. This method is based on a representation of the state space, encompassing all the possible system states and transition relationships. By analyzing the state space, potential security vulnerabilities and attack paths in the system can be identified.

Model checking offers several advantages over simulation and testing methods. Firstly, it provides a comprehensive coverage of all system states and behaviors, enabling the discovery of hidden security vulnerabilities and potential attack paths. Secondly, model checking can provide rigorous mathematical proofs and formal guarantees, allowing for the verification of protocol correctness and satisfaction of security properties. Lastly, model checking can be performed using automated tools, effectively reducing human errors and subjective judgments.

This study adopts an improved CPN-based approach, employing a more granular modeling and control methodology to analyze protocol security. Through formal state space analysis, we can identify security vulnerabilities in the protocol and extract potential attack paths, providing a strong basis for further enhancing and strengthening the ISA100.11a protocol.

The comparison between CPNs and several popular automatic protocol security verification tools is as follows.

ProVerif claims to be able to compute multiple attack paths, using a logic programming approach. However, the computed attack paths are limited, often including only one path [13].

The set of limited attack paths computed by ProVerif is significantly smaller than the set of attack paths extracted by CPN-based methods.

Scyther is a high-performance protocol model verification tool that can provide calculations and an analysis of multiple attack paths. However, it uses a uniform algorithm, attempting to provide a state space analysis for all security protocols in the same manner.

While this approach can identify some attack paths, it falls short of being comprehensive or tailored to specific protocols.

Therefore, the aforementioned or similar automatic verification tools do not require modelers to delve into their internal mechanisms; instead, verification can be achieved by writing scripts in specified formats. In contrast, the high degree of freedom in the CPN modeling process becomes one of its advantages. The state space is entirely controlled by the modeler, allowing for tailored modeling and analysis methods for different protocols. This is often why CPN protocol verification is more effective than automatic protocol verification tools [13].

In conclusion, this study not only addresses the research gap in the security analysis of the ISA100.11a protocol but also introduces a formal analysis approach that offers a new perspective and method for enhancing the security of the wireless communication protocols widely used in industrial control systems. Future work can focus on further refining and expanding this approach to improve the security and reliability of the ISA100.11a protocol.

3. Preliminary Knowledge

3.1. The Communication Process of the ISA100.11a

The ISA100.11a protocol defines a communication pattern that includes an authentication phase and a session phase. In the protocol message flow, a 64-bit chip EUI is used as the device identifier. During the authentication phase, the server can obtain the sender's EUI and perform identity authentication using a shared key. The default group cipher is AES-128, with a 16-bit block size and a 16-bit key size, and it may have hardware encryption support in most devices [6]. The default stream cipher mode should be a CCM* mode, which can be used for encryption only, decryption only, authentication only, extended encryption with authentication, or extended authentication with decryption. Devices encrypt the packets using symmetric key algorithms before sending them. Upon receiving the packets, the devices decrypt them using the same symmetric key algorithm and forward the data to the upper layers. Message integrity verification is performed using the MIC algorithm. The authentication process involves encrypting the messages using a 128-bit session key provided by the security manager. The specific protocol is as follows [1–4]:

When a new device joins the network, it first sends a join request to the nearest announcement router device, requesting forwarding. The router device responds to the request and informs the new device that it can forward the join request.

The new device sends secure, join, and session requests to the announcement router device, including a 128-bit session key and device identification information. The announcement router device forwards the request to the system manager. The PSMO (Public Security Management Object) in the system manager, which should be used by the announcement router, forwards the join request to the security manager and verifies the request information. If the verification is successful, the PSMO sends a response that includes the session key and the encryption policy for the data link layer and session.

After receiving the response, the new device sends a security confirmation request to the PSMO. The PSMO forwards the request to the security manager for confirmation and responds accordingly. The purpose of the join security confirmation is to inform the security manager that the device can recover the master key using the session key, providing evidence that the device possessing the session key is active. If the device fails to provide the session key, the join request will be rejected.

Once the connection is established, the device can engage in secure communication. Before transmitting data, the device divides the overall information into different fields. The communication device sends the session key, data integrity verification code (MIC), and the connected EUI to the central manager. The central manager validates the received information and forwards it to the executing device. Upon receiving the command information, the executing device performs a one-to-one session security key verification and MIC data integrity verification. If the verification is successful, the corresponding command operation is executed, and an execution success message is returned. Otherwise,

the requested command will be rejected. The specific communication mode is depicted in Figure 2.

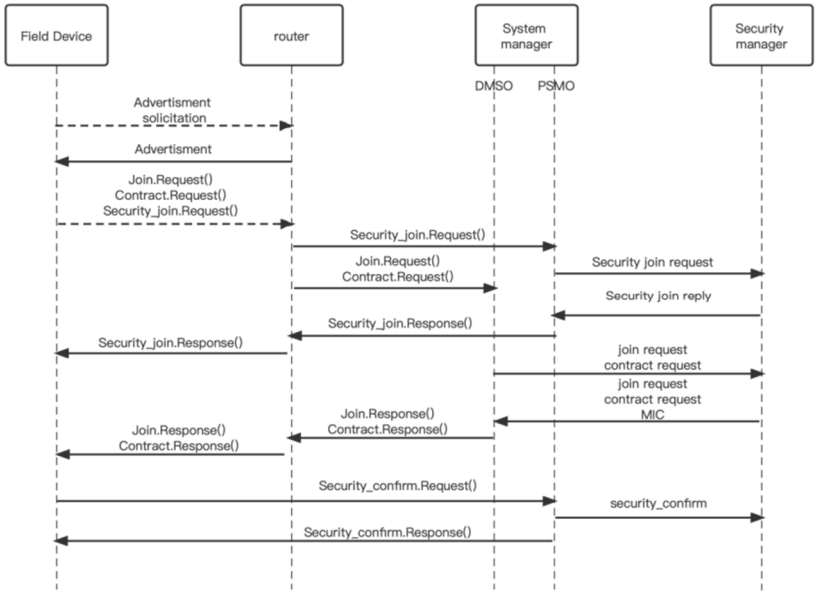

Figure 2. Data flow diagram.

3.2. The CPN Modeling Tool

CPN (Colored Petri Net) is a modeling approach based on Petri net theory that enables the description of concurrent, distributed, and asynchronous systems. It provides an intuitive and formal modeling method that facilitates a clear representation of the system's behavior and structure. In the context of the formal security analysis of the ISA100.11a standard protocol, CPN serves as a modeling tool for analyzing and verifying the protocol's security properties [14–18].

The CPN modeling tool consists of two main components: the tool software and the modeling language.

Tool Software: Commonly used CPN tool software includes CPN Tools4.0 and Design/CPN4.0. This software provides intuitive and user-friendly interfaces and tools that assist users in rapidly constructing and analyzing Petri net models.

Modeling Language: The CPN modeling language encompasses the basic elements of Petri nets, such as the places, transitions, arcs, as well as specialized elements of Colored Petri Nets, such as the color sets, functions, macro definitions, etc. These elements can be used to describe the system's states, events, transitions, and support modularization and hierarchical features during the modeling process.

In the formal security analysis of the ISA100.11a standard protocol, using CPN modeling tools can help analysts construct models in a visual and intuitive interface. These models can accurately reflect the behavior and structure of the protocol, facilitating the formal security analysis and verification. Additionally, CPN modeling tools can assist analysts in simulation and testing, enabling a better understanding and evaluation of the protocol's security properties [19–22].

3.3. Attacker Model

In the formal security analysis of the ISA100.11a standard protocol, the Dolev-Yao attacker model is used. This model is a widely adopted attacker model that is commonly employed for the formal analysis of security protocols.

The Dolev-Yao model exhibits the following characteristics:

Complete Observability: In the Dolev-Yao model, attackers possess the complete observability of all messages on the network, including both content and traffic. This

corresponds to the capabilities of a man-in-the-middle (MITM) attack, where attackers can intercept and monitor communications.

Complete Controllability: The model assumes that attackers have full control over network communication, enabling them to actively intervene in communication by modifying, replaying, injecting, or blocking messages. This aligns with attacks such as identity authentication attacks and access point deception.

Complete Knowledge of Protocols and Algorithms: The Dolev-Yao attacker model assumes that attackers possess comprehensive knowledge of all details related to the encryption algorithms, protocol design, and communication protocols in use. This corresponds to attackers having sufficient knowledge to understand and analyze the protocols, including the encrypted content.

Through these features, the Dolev-Yao model provides a means for protocol designers to evaluate the security of protocols under worst-case scenarios. Although it is a theoretical abstraction, it aids in comprehending and addressing the security of protocols when confronted with rational and powerful attackers. The Dolev-Yao attacker model assumes that attackers can intercept, manipulate, and inject messages in the communication, as well as break cryptographic algorithms and forge digital signatures. While this attacker model has certain theoretical limitations, it serves as an abstract model that helps analysts evaluate the strength and robustness of security protocols [23–25].

In the Dolev-Yao attacker model for the ISA100.11a standard protocol, attackers can launch attacks on the protocol through the following means:

Eavesdropping: Attackers can gain access to sensitive information by eavesdropping on the communication within the protocol.

Replay Attacks: Attackers can record and replay messages exchanged during the communication process, leading to duplicate or erroneous operations within the protocol.

Tampering: Attackers can modify the content of the communication, thereby influencing the execution of the protocol.

Spoofing: Attackers can forge messages to deceive the system into performing incorrect operations.

Denial of Service (DoS) Attacks: Attackers can consume system resources by continuously sending invalid messages, causing the system to malfunction or become unresponsive [26–34].

In summary, the Dolev-Yao attacker model provides a commonly used security model that assists analysts in evaluating the strength and robustness of security protocols, effectively safeguarding the security and stability of communication systems.

4. ISA100.11a Protocol Modeling

4.1. Definition of the Color Set

During the analysis of the interaction process for protocol session establishment and data transmission, several details are considered. In the session establishment process, the device needs to broadcast and include the device's Connection ID, which is also known as the EUI, as well as a unique Session Connection ID. After a successful connection establishment, there are messages for successful validation and messages for resetting operations after a connection failure. In the research process, individual basic information elements are listed, and they are combined to form the session message format. After a successful connection establishment, the transmitted secure data packets include the basic data, MIC (Message Integrity Check), command information, session ID, and logging information. Communication between the master and slave stations involves the successful execution of commands, which is followed by a reply containing success counters, as well as error messages in case the command execution fails. The main definitions of color sets are presented in Table 1 and more are in Figure 3.

Table 1. Definition of color set.

Key Elements	Definition of Color Set
EUI	colset EUI = int;
JoinKey	colset JoinKey = string;
SessionKey	colset SessionKey = string;
Challenge	colset Challenge = bool;
Payload	colset Payload = string;
Mic	Coleset Mic = string;
Req1	colset Req1 = product EUI*JoinKey*JoinData;
Req2	colset Req2 = product NewKey*Challenge*EUI*SecurityPolicy*INT*Payload*Mic;
Res1	colset Res1 = product Policy*SessionKey*Challenge*NewKey;
SuccessRes	colset SuccessRes = product Challenge*SecurityPolicy*Payload;
TransMes	colset TransMes = product Newkey*Challenge*EUI*SecurityPolicy*INT;
PayLoadMic	colset PayLoadMic = product Payload*Mic

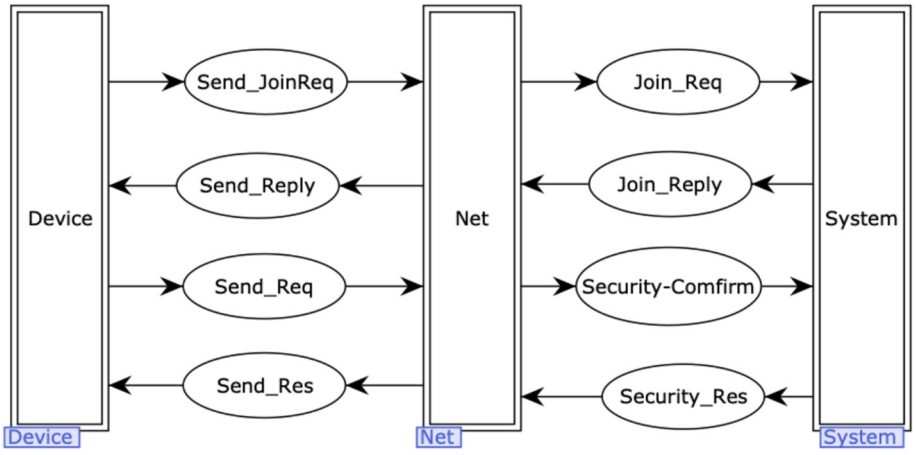

Figure 3. Top-level model of the protocol.

4.2. Formal Analysis of the ISA100.11a

The establishment process of this protocol follows a top-down approach for formal modeling. The network is divided into three components: the sender, the receiver, and the central manager. These components are further simplified into three levels: top, middle, and bottom. To reduce model complexity and better represent the network communication process, each level is subdivided and the message flow of the protocol is described in detail. In the model, ellipses represent places, rectangles represent transitions, and double line rectangular transitions represent alternative transitions, indicating that these transitions contain more detailed sub-models.

The top-level model of the protocol consists of three alternative transitions and eight places. It abstractly represents the communication process between the terminal device and the server. The specific details are illustrated in Figure 3.

The middle-level model of the protocol consists of four alternative transitions and nine places. The specific process of the sender sending a join request message to the receiver is represented by the alternative transition "Connection". The process of the receiver receiving the message and analyzing it is represented by the alternative transition "Security manager". The process of the sender initiating the request and the receiver replying to the request is represented by the alternative transition "Security_commit". The process of the receiver receiving the request and sending data to the sender is also included in the alternative transition "Security manager". The specific details are illustrated in Figure 4.

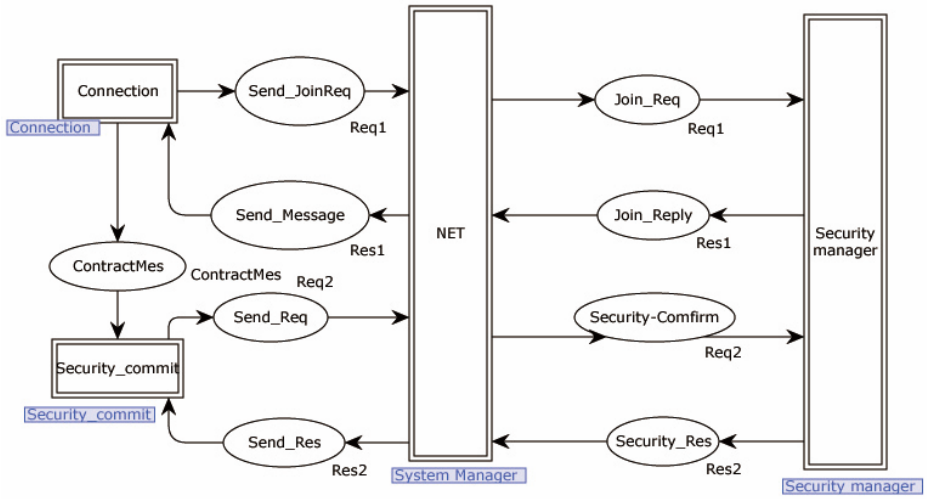

Figure 4. Middle-level model of the protocol.

Figure 5 represents the internal model of the alternative transition "Connection". The "Encrypt" transition is responsible for encrypting the request data using the joining key. The "Key" and "Data" places store the main key and data of the sender, respectively. The "Device EUI" place holds the unique EUI identification code of the device. The encrypted data are stored in the "Join_Message" place and then sent to the "Commit_Mes" transition. The "Req" place forwards the information to the "Send_Req" transition, which subsequently transmits it to the network (NET).

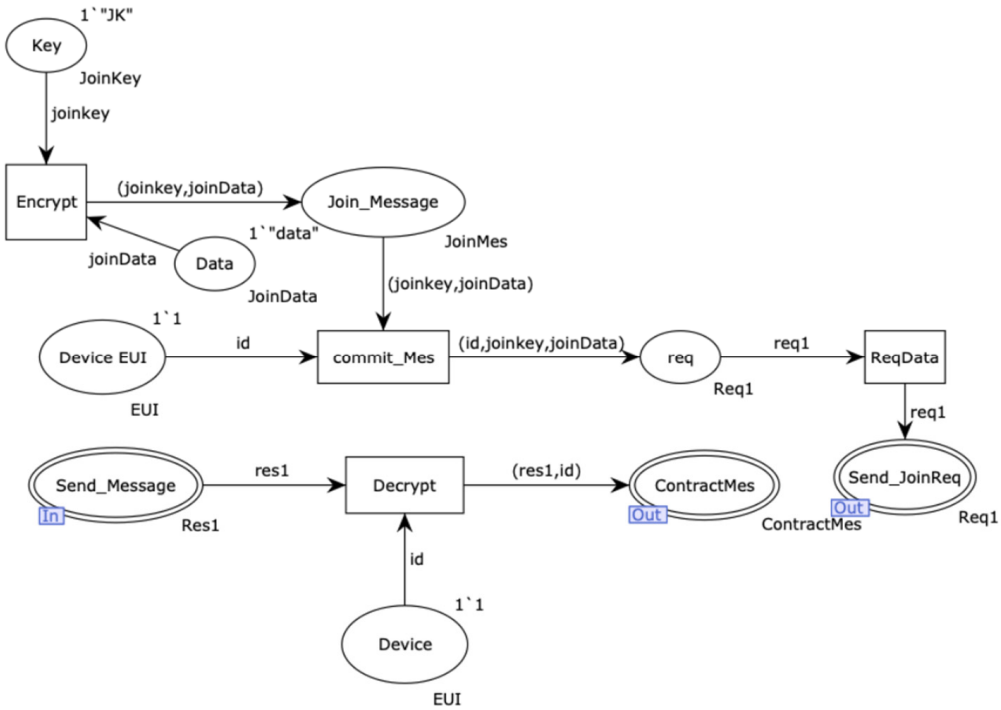

Figure 5. Internal model of the alternative transition "Connection".

Figure 6 illustrates the internal model of the alternative transition "SecurityManager" for receiving join requests. Firstly, the "DecryptMesStation" transition decrypts the received message using the public join key. Next, the decrypted message containing the EUI, joinData, and joinKey is sent to the "EUI", "Data", and "Key" places, respectively. The "VerifyID" transition validates the legality of the EUI. The "Produce newkey" transition verifies the join key and generates a new main key, which is sent to the "NewKey" place.

The "Produce Challenge" transition verifies the join information and generates a challenge to determine the server's availability. Finally, the "EncryptResMes" transition encrypts all the data to be sent and transmits them to the device via the network (NET).

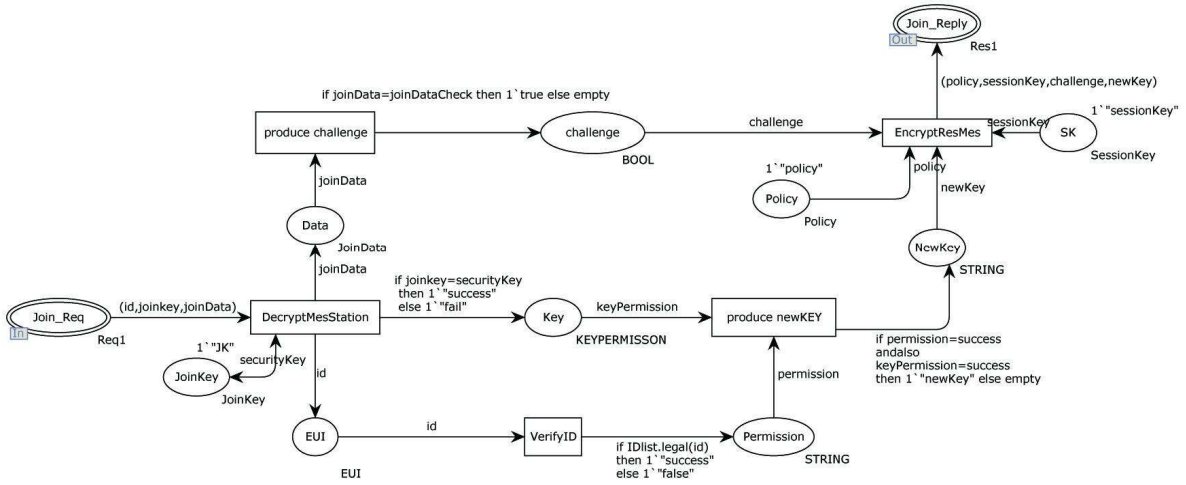

Figure 6. Internal model of the alternative transition "SecurityManager" (1).

Figure 7 depicts an internal model of the alternative transition "Security_commit" for sending session requests after receiving a join request reply. The "Decrypt" transition decrypts the received message using the public join key. The "VerifyChallenge" transition verifies the challenge to check the server's status. If the verification is successful, a new challenge is sent to the next place. The "Security_commit" transition encrypts all the data to be sent and transmits them to the executing device via the network (NET).

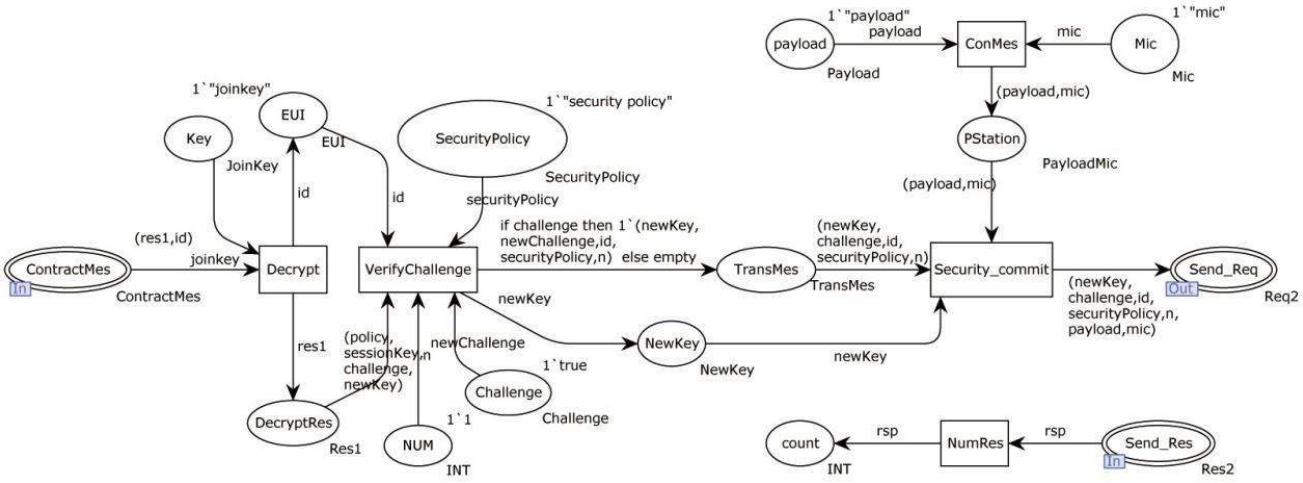

Figure 7. Internal model of the alternative transition "Security_commit".

Figure 8 illustrates an internal model of the alternative transition "SecurityManager" for receiving and processing data sent by the device. The "Decrypt" transition uses the master key to decrypt the received message. Next, the "Verify MIC" transition checks the integrity of the information, followed by the "Verify challenge" transition to verify if the device is alive. Subsequently, the "Verify key" transition validates the legitimacy of the key. If all the verifications succeed, the session is considered complete.

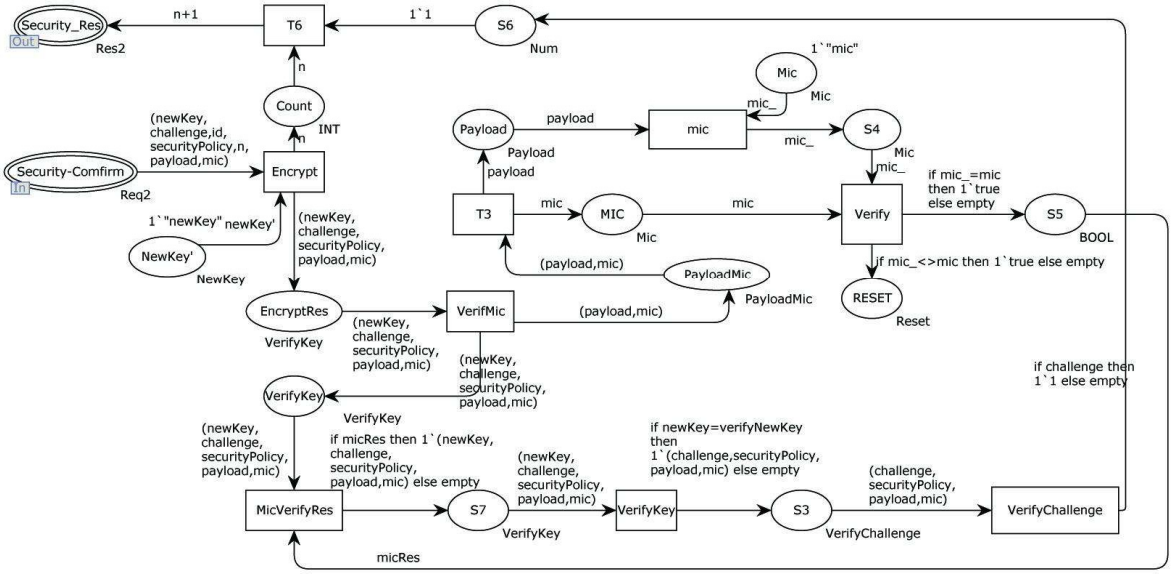

Figure 8. Internal model of the alternative transition "SecurityManager" (2).

4.3. Protocol Consistency Analysis

A consistency analysis was performed on the ISA100.11a protocol using the state space analysis tool provided by CPN Tools. By analyzing the data in Table 2, it can be observed that the number of the reachable state nodes, directed arcs, strongly connected nodes, and strongly connected arcs is the same. This indicates that the established protocol model does not lead to state loops, and that all state nodes are reachable. In addition, the number of dead nodes is 1, which indicates that all requests are executable and that the endpoint of the protocol's session interaction is uniquely determined in any case. At the same time, there are no dead or live transitions, indicating that there are no nodes that cannot be reached and no nodes that are always in an active state, and the established model can run correctly.

Table 2. State space analysis result of the original model.

Type	Number
State Space Node	33
State Space Arcs	37
Scc Graph	33
Arcs	37
Dead Markings	1
Dead Transitions	0

4.4. Protocol Security Assessment

In the network transport layer of the ISA100.11a protocol model, attackers can exploit replay attacks, tampering attacks, and known key leakage impersonation attacks to compromise communication. Attackers can analyze whether the messages are sent by the same device by intercepting the device's identity information. Once the attacker obtains the join key, they can decrypt the initial transmission between the device and the server and tamper with the transmitted information.

4.4.1. Known Key Leakage Impersonation Attack Model

In Figure 9, blue components represent the modeling of known key leakage impersonation. For example, Transition T0 captures the protocol transmission, and Places P0, P1, P2, and P3 store the decomposed and to-be-decomposed messages. Transitions T1 and T2 represent the decrypting and reassembling of the intercepted messages using the known key.

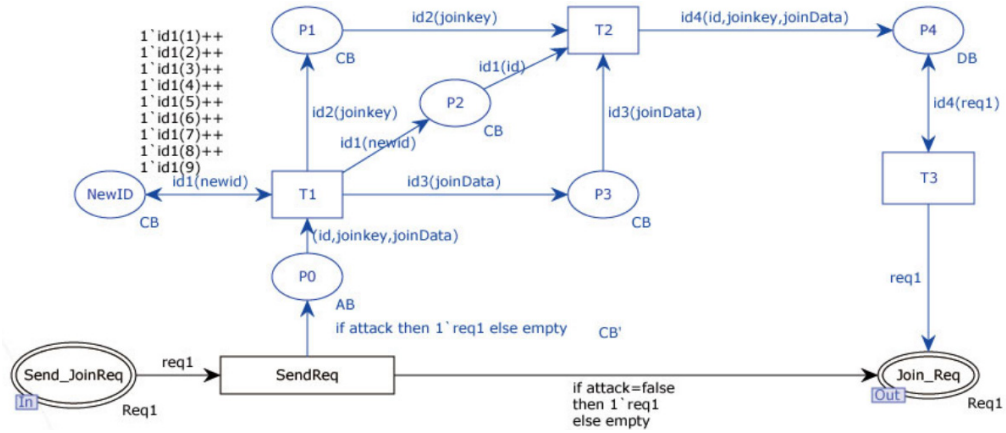

Figure 9. The original protocol attacker model (a).

4.4.2. Tampering Attack Model

In Figure 10, the blue components represent the modeling of the tampering attacks.

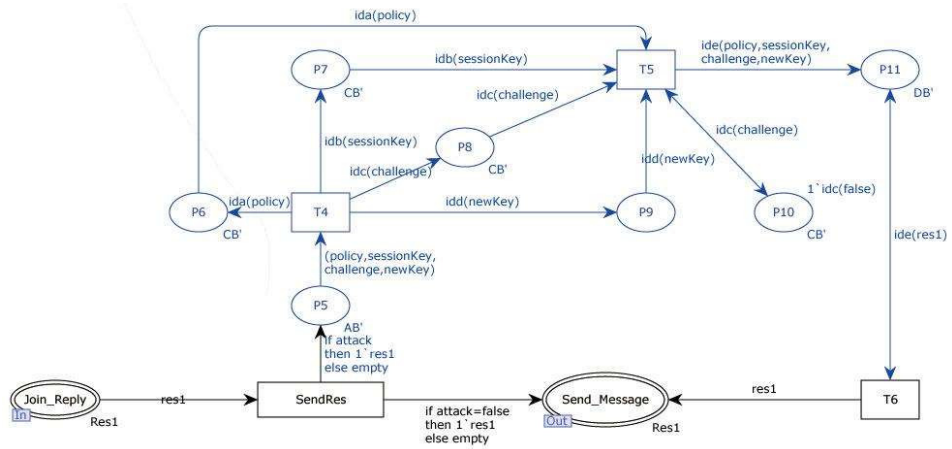

Figure 10. The original protocol attacker model (b).

4.4.3. Man-in-the-Middle Attack Attack Model

In Figure 11, the blue components represent the modeling of man-in-the-middle attack. The red components in Figure 11 represent the modeling of replay attacks.

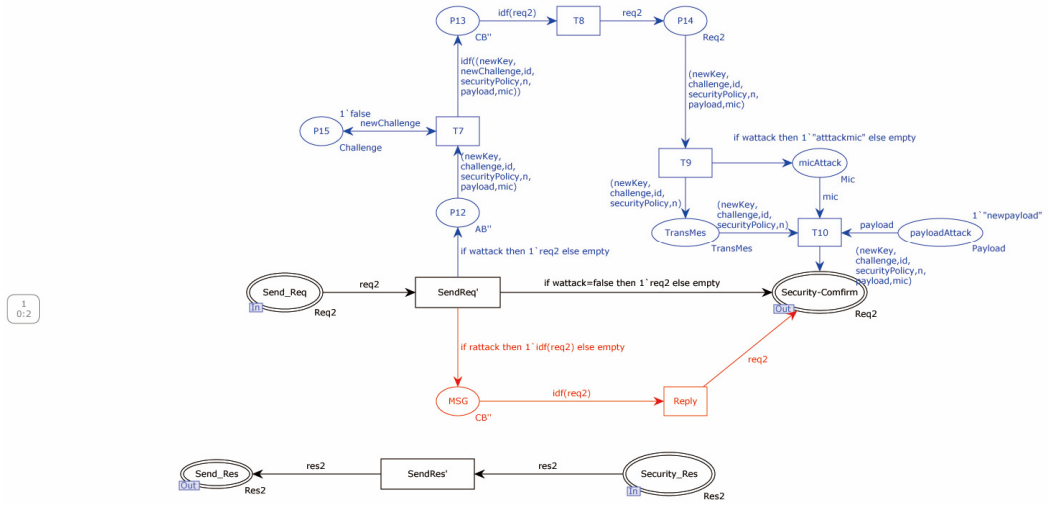

Figure 11. The original protocol attacker model (c).

The red components in Figure 11 represent the modeling of a replay attack. The attack is divided into two stages: message decomposition and message synthesis. When the attacker intercepts a message during the session, they first decompose the message and then synthesize a new message from the decomposed parts. It is important to complete all decomposition operations before the synthesis. The attacker's messages are stored using multisets, where the AB multiset stores messages to be decomposed and the CB multiset stores the atomic messages generated during the decomposition. The DB multiset stores the synthesized messages. The decomposition and synthesis process involves transferring messages between the multisets according to specific rules. Only the synthesized messages in the DB multiset can be sent over the communication channel. In the specific application steps, decomposition rules are applied before synthesis rules to avoid cyclic operations.

By modeling these attack scenarios and simulating the decomposition and synthesis processes, the protocol's vulnerability to replay attacks, tampering attacks, and known key leakage impersonation attacks can be evaluated. This allows for the identification of potential security weaknesses and the development of countermeasures to enhance the security of the protocol.

4.5. ISA100.11a Protocol Security Property Verification Analysis

Through protocol state space analysis, it can be observed that the ISA100.11a protocol effectively prevents man-in-the-middle tampering attacks. During protocol execution, the device verifies the integrity of transmitted information by checking the Message Integrity Code (MIC). If an attacker tampers with the information and causes the MIC verification to fail, the protocol takes appropriate actions to mitigate the impact of tampering attacks. The state space analysis results, as shown in Table 3, confirm this behavior. Additionally, the value of the "RESET" place in the analysis result (as indicated in Table 4) is "true", indicating that the attack was unsuccessful.

Table 3. State space analysis result.

Type	Disclosure Attack	Reply Attack	Modify Attack
State Space Nodes	38,230	57	43
State Space Arcs	138,156	86	61
Scc Graph Nodes	38,230	57	43
Scc Graph Arcs	138,156	86	61
Dead Markings	30	1	1
Dead Transitions	18	10	0

Table 4. The value of some places in the state space report.

Place	Value
Security_manager'RESET 1	1`true
Security_commit'DecryptRes 1	1`("policy", "sessionKey", false, "newKey")
System_Manager'NewID 1	1`id1(1)++
Security_manager'Policy 1	empty

However, the ISA100.11a protocol is vulnerable to known key compromise and device ID spoofing attacks, as well as replay attacks. From the state space analysis report, it can be observed that an attacker, upon obtaining the device's join key, can effectively acquire the session key used in the communication process, forge a legitimate EUI, and tamper with information to act as a man-in-the-middle, sending requests to the executing device. As a result, the attacker can gain access to all the secret information exchanged between the communicating devices. Additionally, the protocol does not ensure the anonymity of the communication devices since the EUI of the sender and receiver remains unchanged during the communication process.

5. Enhanced Plan Utilizing ISA100.11a Protocol

By introducing an attack model to the original protocol model and analyzing the runtime states and state space report, it was discovered that the protocol is vulnerable to man-in-the-middle attacks, including replay and tampering attacks. The communication devices are not anonymized, making it easy for attackers to forge an EUI and launch attacks. To address the identified security vulnerabilities, reinforcement measures can be applied during the session establishment authentication and secure data transmission processes.

To enhance the security validation between the communication devices and executing devices, a point-to-point secure authentication can be implemented during the session establishment, replacing the original protocol's reliance solely on the network's central manager for validation. Additionally, random number-generated hash values and timestamps can be introduced during secure data transmission and session processes to improve the security and resistance against replay attacks.

While incorporating these reinforcement measures, the basic security functionalities of the protocol are retained. After the central manager validates a device as a legitimate one, if the device is hijacked, any malicious activity conducted by the hijacked device would be detected by other nodes or the central manager. Once illegal attack behavior is detected, the hijacked device will be excluded from the list of legitimate devices by the central manager, thereby mitigating the security threats posed by hijacked validated devices.

To anonymize device EUIs, anonymous numbering can be used, and the server can perform an internal lookup and comparison based on the assigned numbers instead of the actual EUIs, ensuring the anonymity in the communication process.

5.1. Improved Protocol Communication Process

The specific communication process of the protocol is shown below, and Table 5 lists important symbols.

Table 5. Communication Process Symbol Representation.

Symbol	Definition
EUI_{device}	The ID of device
EUI_{center}	The ID of center
EUI_{server}	The ID of server
$H(.)$	One-way hash function
$\oplus$	XOR operation
$\|\|$	Concatenation operation
SK	Session key
CK	Communication key

The communication device sends a connection authentication request to the network central manager, selects a random number R1, calculates $\alpha = H(R1)$, $x = H(EUI_{device} + R1)$, and then sends (JK, α, x).

The network central manager processes the received connection information, looks up the stored user identification hash function group, calculates $x' = H(EUI_{device}) \times \alpha$, and determines $x = x'$. If it exists, the communication device is authenticated successfully, otherwise the authentication fails. After authentication, the manager selects a random number R2, and calculates $\beta = H(R2), y = H(EUI_{center}) \times \alpha \times \beta$. The connection request information is combined (α, β, y) and sent to the execution device.

The execution device receives the communication device's connection request information from the network central manager, looks up the stored central manager identification hash, calculates $y' = H(EUI_{center}) \times \alpha \times \beta$, and determines $y = y'$. If equal, the validation is successful. It selects a random number R3, calculates $\gamma = H(R3)$, $z = H(EUI_{server}) \times \gamma \times \beta)$ and finally sends the connection response result (z, γ) to the network central manager.

After receiving the response information sent by the execution device, the network central manager looks up the stored central manager identification hash, calculates $z' = H(EUI_{server}) \times \gamma \times \beta)$, and determines $z = z'$. If equal, the validation is successful. The network central manager generates a communication key, calculates $K_1 = CK \oplus \alpha$, $K_2 = (CK||EUI_{device}) \oplus \gamma$ and finally sends K_1 to the communication device and sends K_2 to the execution device.

After receiving the message, the communication device uses α XOR K_1 to obtain the communication key. After receiving the message, the execution device uses γ XOR K_2 to obtain the communication key and EUI_{device}.

The execution device generates a session key, calculates $M_1 = E(CK, SK)$, and sends it to the communication device.

After receiving the message, the communication device decrypts it using the communication key to obtain the session key SK. The process of the session connection and execution command information interaction ends here.

5.2. The Model of the Improved Protocol

In the improved middle-layer model, a new transition called "Center" is added to represent a third-party network's central manager. This transition is responsible for performing hash verification and obtaining the communication and session keys. Additionally, 12 new places are introduced to facilitate the authentication of identities, session establishment, and the interaction process for executing commands before formal communication begins, as shown in Figure 12.

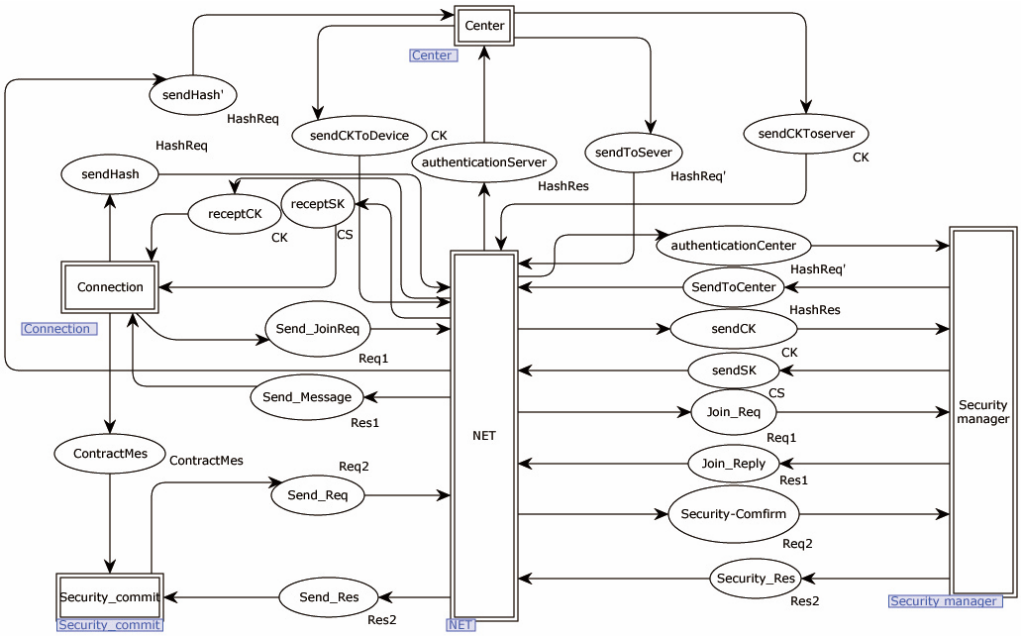

Figure 12. The middle model of the new scheme.

Figure 13 illustrates the detailed components of the alternative transition "Connection". It encompasses the communication device's transmission of a connection authentication request to the network central manager. The "Random" place generates a random number, which is then sent to the "Hash" transition. The "Hash" transition computes the values x, represented as hashR and hashER in the diagram. Subsequently, the transition "T0" transmits (hashER, hashR, DeviceID, time') to "NET", which forwards it to "Center".

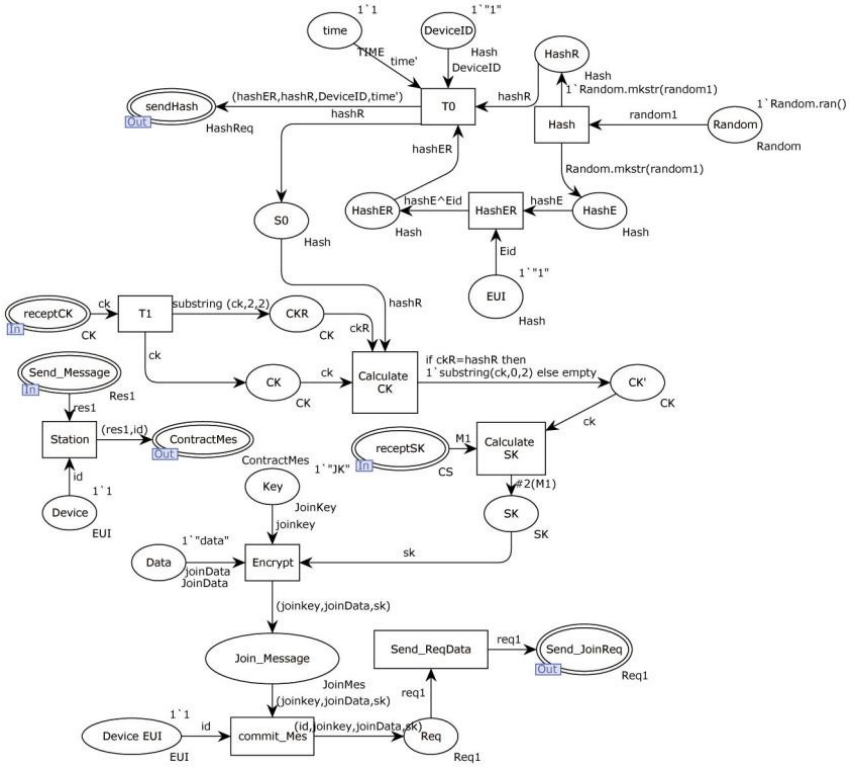

Figure 13. Alternative transition "Connection" internal model.

Furthermore, the diagram includes the successful authentication of both parties. After the "T1" transition receives the encrypted communication key CK, it proceeds to verify the consistency of the hash for random number R1. Then, the "Calculate CK" transition computes the value of CK. Subsequently, the "Calculate SK" transition utilizes CK to calculate the session key SK. The resulting session key SK is stored in the place "SK".

Figure 14 represents the internal model of the alternative transition "Center". It involves the validation of the communication device's connection authentication request. The "Time" place stores the previous timestamp and compares it with the current request. The "verifyDevice" transition verifies the legitimacy of the communication device and the timestamp. Upon its successful verification, a random number R2 is generated.

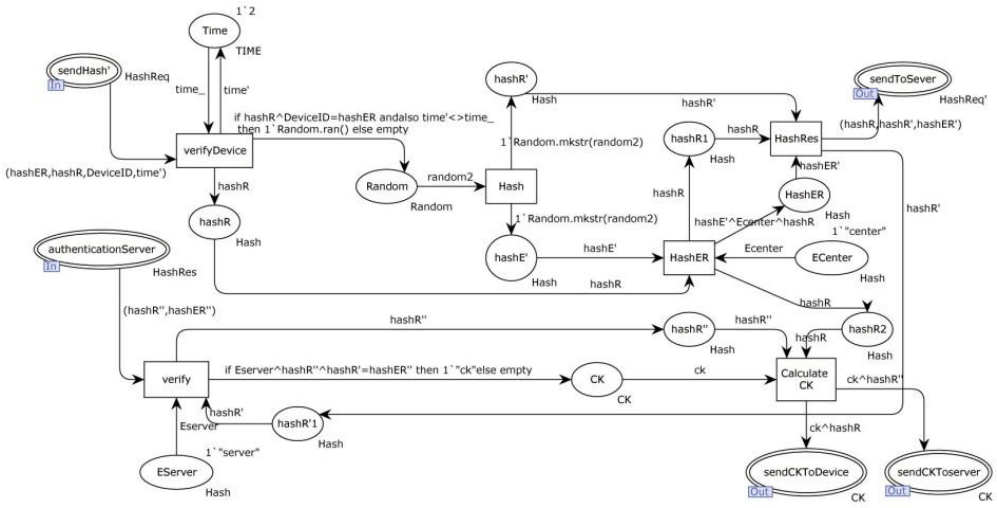

Figure 14. Alternative transition "Center" internal model.

The "Hash" transition calculates the values hashR' and hashER', corresponding to the diagram. Subsequently, the "HashRes" transition sends (hashR, hashR', hashER') to "NET", which forwards it to the "Server". The "authenticationServer" transition receives the hash sent by the executing device and then the "verify" transition verifies the hash of the executing device. After its successful verification, the communication key CK is generated.

Next, the "Calculate CK" transition computes the values of K1 and K2. K1 is sent to the communication device, and K2 is sent to the executing device.

Figure 15 represents the additional model within the alternative transition "SecurityManager". It involves the validation of the connection authentication request from the network management center "Cener". The "verify" transition verifies the legitimacy of the network management center "Center". Once the validation is successful, a random number R3 is generated.

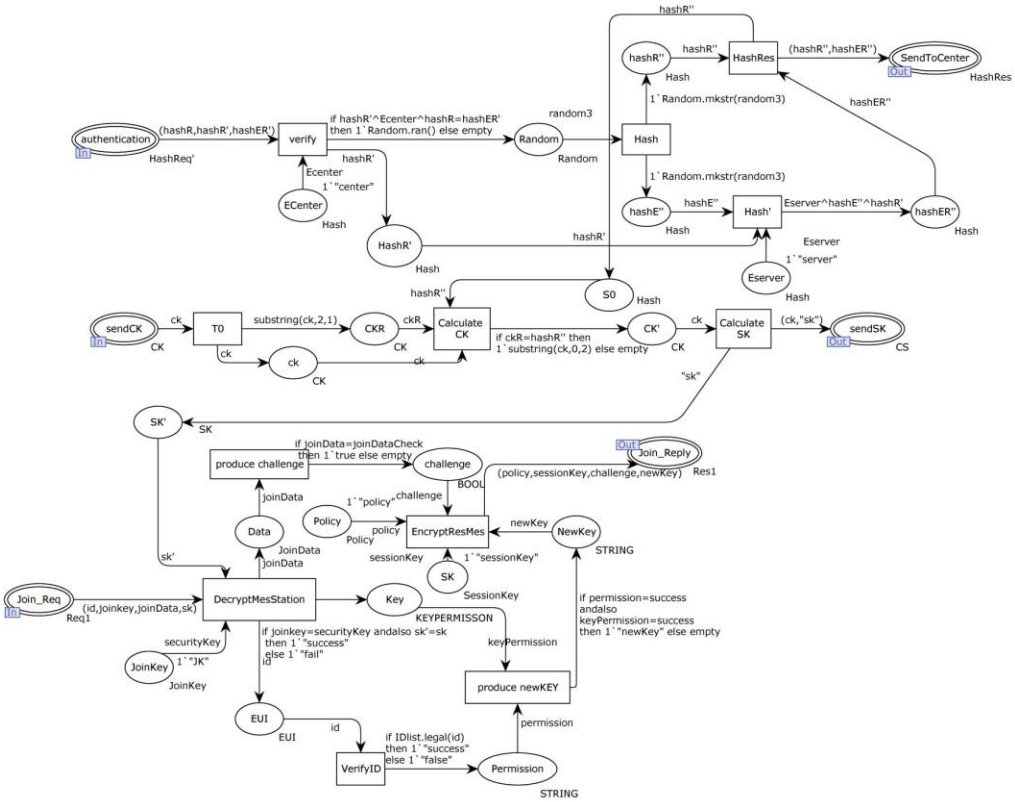

Figure 15. "SecurityManager" transition "Center" internal model.

The "Hash" transition calculates the values hashR'' and hashER'', as shown in the diagram. Subsequently, the "HashRes" transition sends (hashR'', hashER'') to the "NET" place, which further forwards it to the "Center". The "T0" transition receives the encrypted communication key CK. Then, it verifies the hash of the random number R3, followed by the "Calculate CK" transition that computes the value of CK.

Furthermore, the "Calculate SK" transition uses the value of CK to calculate the SK value. The encrypted SK is then sent to the communication device.

5.3. Improved Protocol Security Assessment

In the improved protocol, the Dolev-Yao attacker model is reintroduced to simulate network-level man-in-the-middle attacks, including tampering, replay, and known key attacks. The red portion represents replay attacks, the blue portion represents tampering attacks, and the purple portion represents known key attacks. The specific details are depicted in Figure 16.

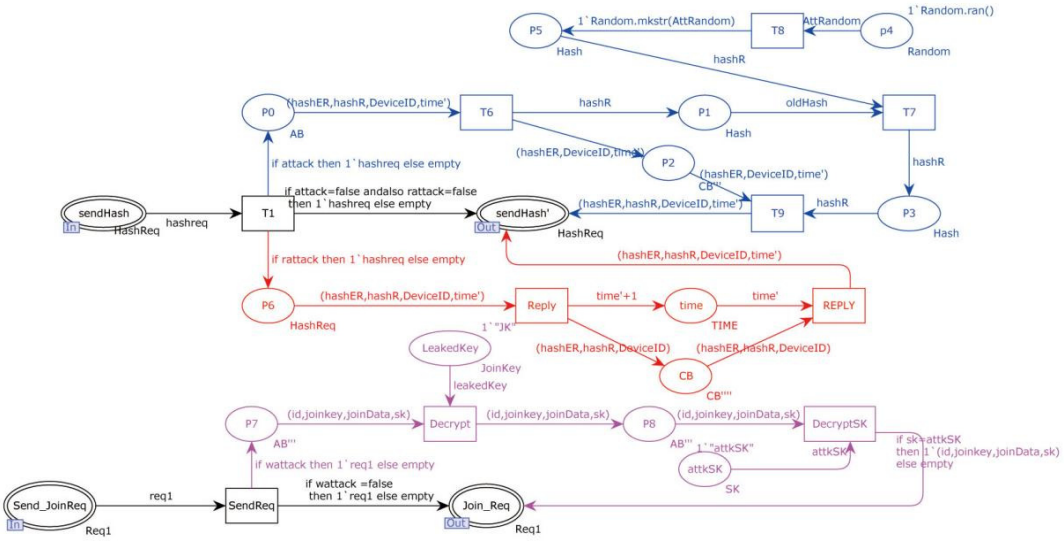

Figure 16. The improved attacker models of the protocol.

Table 6 compares the state space reports of the improved protocol and the protocol with the introduced attacker model. Prior to the inclusion of the attacker model, the state space report of the improved protocol showed one dead node and zero dead transitions, indicating normal operation. After introducing tampering attacks, since all hashes are constructed using their respective random numbers, the attacker cannot obtain the values of the random numbers, making it impossible to crack the hash values. After tampering, the center fails the hash verification and authentication. The 46 dead transitions are caused by authentication failures. Due to the inclusion of timestamps in the requests, replay attacks are also ineffective, and the 49 dead transitions are a result of the server detecting replay attacks. By introducing known key attacks in the protocol, the key used in the broadcast routing request can be obtained. However, the session key used for encryption is private and not transmitted. Both the communication device and the execution device independently calculate their own session keys after authentication, making it impossible for the attacker to crack the encrypted messages protected by the session key. The 27 dead transitions are a result of the attacker's inability to decrypt the encrypted information.

Table 6. Comparison of state space reports.

Type	Improved Protocol	Modify Attack	Reply Attack	KCIA Attack
State Space Nodes	166	30	32	398
State Space Arcs	272	53	60	807
Scc Graph Nodes	166	30	32	398
Scc Graph Arcs	272	53	60	807
Dead Markings	1	1	1	2
Dead Transitions	0	46	49	27

5.4. Security Analysis of the Improved Protocol

The following is a security analysis of the improved protocol against various common attacks.

5.4.1. To Mitigate Replay Attacks

The sender and receiver include a hash value and a timestamp when communicating with the server. When a device receives a message, it first checks if the timestamp is present. If it is, the device will discard the message immediately. This ensures that any messages with a timestamp from a previous session or transmission are rejected, effectively preventing replay attacks.

5.4.2. To Counter Impersonation Attacks

Attackers attempt to initiate sessions with other devices using intercepted device identity information. However, in the improved protocol, the device's EUI is not disclosed before communication, and after authentication, each communication is encrypted using a session key that the attacker cannot obtain. As a result, the attacker is unable to acquire the genuine EUI of the device, thereby preventing impersonation attacks.

5.4.3. To Counter Tampering Attacks

Where attackers intercept plaintext messages and modify them before sending them to the recipient, the improved protocol employs several measures. During the authentication process, the sender and recipient exchange hash values of random numbers. Even if an attacker intercepts and alters the hash value of a message, upon receiving the message, the communicating parties compare the received hash value with their own stored hash value. If a mismatch is detected, the message is discarded. During the communication phase, the sender utilizes a session key to encrypt the calculated Message Integrity Code (MIC). Without aknowledge of the session key, attackers are unable to tamper with the message or bypass the integrity verification provided by the MIC.

5.4.4. To Mitigate the Risk of Key Leakage and Masquerade Attacks

The improved protocol adopts several measures. The communication key is securely distributed by a trusted third-party server to the communicating parties. Before each communication session, both parties retrieve the communication key from the trusted server. Even if an attacker manages to compromise the communication key used in a specific session, they cannot decrypt the content of messages transmitted before or after that session. This ensures the confidentiality and freshness of the communication key. Furthermore, even if a subsequent key is compromised, without knowledge of the session key, attackers are still unable to decipher the transmitted messages. This approach provides end-to-end security, safeguarding the confidentiality of the communication even in the event of key leakage.

5.4.5. Anonymity

In the improved protocol, the anonymity of the devices is achieved through the following steps. During the identity authentication phase, the sender must obtain the hashed value of the receiver's identity EUI from a trusted third party. Additionally, during communication between the sender and receiver, the sender utilizes the hashed value of the receiver's identity EUI for message transmission. As a result, the communication process ensures the anonymity of device information, as the actual identity EUI is not directly revealed in the messages exchanged between the sender and the receiver.

5.4.6. Conclusions

Table 7 provides a security analysis comparison between the original protocol and the improved protocol, addressing issues such as tampering, replay attacks, and anonymity. The data in the table clearly demonstrate that the improved protocol possesses stronger security (In this table, $\surd$ indicates support and $\times$ indicates no support).

Table 7. A comparison of the security between the original and improved protocols.

Attack Protocols	ISA100.11a	Improved Protocol
Tamper-resistant attack	$\surd$	$\surd$
Resist replay attacks	$\times$	$\surd$
Anti-simulated attack	$\times$	$\surd$
Resist known key attacks	$\times$	$\surd$
Anonymity	$\times$	$\surd$

In order to address the security vulnerabilities present in the ISA100.11a protocol, we have proposed a novel and enhanced protocol. This protocol introduces a trusted third-party cloud-based server, which generates the communication keys and securely transmits them to the communicating entities, thereby ensuring the protocol's resistance against key leakage and impersonation attacks. Moreover, the improved protocol adopts the utilization of random number hashing, thereby ensuring both security and resource efficiency without imposing significant computational burdens on the original devices. Additionally, the enhanced protocol guarantees the anonymity of the communication devices, thereby enhancing the security and confidentiality aspects of the protocol.

6. Conclusions

This study focuses on addressing the security vulnerabilities within the ISA100.11a standard protocol, with a specific focus on potential risks tied to malicious attackers intercepting sensitive data transmitted by devices and analyzing captured messages to extract private information. To mitigate these security concerns, we employed the CPN tool to model the ISA100.11a protocol and introduced an attacker model for security analysis. Upon analyzing the original protocol, it was observed that there were 30 dead nodes and 1 dead transition for one attack, and 18 dead nodes and 10 dead transitions for another attack. This revealed vulnerabilities to key leakage and device ID spoofing attacks, as well as replay attacks.

To address these issues, we propose an improved protocol that incorporates a trusted third-party cloud-based server. This server is responsible for generating communication keys and securely transmitting them to the communicating entities, thus ensuring its resistance against key leakage and device ID spoofing attacks. Additionally, the enhanced protocol employs a random number hashing approach, ensuring both its security and efficient operation on lightweight devices. Furthermore, the improved protocol guarantees the anonymity of communication devices, thereby enhancing overall security and confidentiality.

Through modeling the enhanced ISA100.11a protocol and subjecting it to security analysis with the attacker model, we observed that the number of dead nodes in the state space was reduced to 1. This validates the improved protocol's security. It is important to note that this study primarily focuses on enhancing the security aspects of the protocol and may have limitations in considering real-time interaction requirements. Future work should aim to strike a balance between security enhancements and minimizing time overhead to meet real-time constraints.

Author Contributions: Conceptualization, T.F.; methodology, T.C.; formal analysis, T.C.; investigation, T.F.; software, T.C.; data curation, X.G.; writing—original draft, T.C.; writing—review and editing, T.F. and X.G.; visualization, T.C.; validation, T.C.; resources, X.G.; supervision, X.G.; funding acquisition, T.F. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Natural Science Foundation of China (Grant No. 62162039, 61762060) and the Natural Science Foundation of Gansu Province, China (Grant No 23YFGA0060).

Data Availability Statement: The data used to support the findings of this study are included within the article.

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

References

1. Wang, G. Comparison and Evaluation of Industrial Wireless Sensor Network Standards ISA100. 11a and *Wireless*HART. Master's Thesis, Chalmers University of Technology, Gothenburg, Sweden, 2011.
2. Lee, J.; Park, K.; Kim, S. A Study on Cyber Security Threat and Security Requirements for Industrial Wireless Communication Devices. *J. Korea Inst. Inf. Secur. Cryptol.* **2020**, *30*, 757–770.
3. Jeong, J.H.; Kwon, S.M.; Shon, T.S. Security Threats Analysis and Security Requirement for Industrial Wireless Protocols: ISA 100.11 a and WirelessHART. *J. Korea Inst. Inf. Secur. Cryptol.* **2019**, *29*, 1063–1075.
4. Kitano, K.; Yamamoto, S. Strong security measures implemented in isa100. 11a wireless system. *Yokogawa* **2014**, *57*, 2.

5. Rezha, F.P.; Shin, S.Y. Performance analysis of ISA100. 11a under interference from an IEEE 802.11 b wireless network. *IEEE Trans. Ind. Inform.* **2014**, *10*, 919–927. [CrossRef]
6. Raptis, T.P.; Passarella, A.; Conti, M. A survey on industrial Internet with ISA100 wireless. *IEEE Access* **2020**, *8*, 157177–157196. [CrossRef]
7. Zhang, X.; Wei, M.; Wang, P.; Kim, Y. Research and implementation of security mechanism in ISA100. 11a networks. In Proceedings of the 2009 9th International Conference on Electronic Measurement & Instruments, Beijing, China, 16–19 August 2009.
8. Dinh, N.Q.; Kim, D.S. Performance evaluation of priority CSMA-CA mechanism on ISA100. 11a wireless network. *Comput. Stand. Interfaces* **2012**, *34*, 117–123. [CrossRef]
9. Saleem, M.A.; Shamshad, S.; Ahmed, S.; Ghaffar, Z.; Mahmood, K. Security Analysis on "A Secure Three-Factor User Authentication Protocol with Forward Secrecy for Wireless Medical Sensor Network Systems". *IEEE Syst. J.* **2021**, *15*, 5557–5559. [CrossRef]
10. Pu, C.; Zerkle, H.; Wall, A.; Lim, S.; Choo, K.-K.R.; Ahmed, I. A Lightweight and Anonymous Authentication and Key Agreement Protocol for Wireless Body Area Networks. *IEEE Internet Things J.* **2022**, *9*, 21136–21146. [CrossRef]
11. Amoah, R.; Suriadi, S.; Camtepe, S.; Foo, E. Security analysis of the non-aggressive challenge response of the DNP3 protocol using a CPN model. In Proceedings of the 2014 IEEE International Conference on Communications (ICC), Sydney, NSW, Australia, 10–14 June 2014; pp. 827–833. [CrossRef]
12. Bhurke, A.U.; Kazi, F. Methods of Formal Analysis for ICS Protocols and HART—IP CPN modelling. In Proceedings of the 2021 Asian Conference on Innovation in Technology (ASIANCON), Pune, India, 27–29 August 2021; pp. 1–7. [CrossRef]
13. Bernabé-Sánchez, I.; Fernández, A.; Billhardt, H.; Ossowski, S. Problem Detection in the Edge of IoT Applications. *Int. J. Interact. Multimed. Artif. Intell.* **2023**, *8*, 85–97. [CrossRef]
14. Kampourakis, V.; Gkioulos, V.; Katsikas, S. A systematic literature review on wireless security testbeds in the cyber-physical realm. *Comput. Secur.* **2023**, *133*, 103383. [CrossRef]
15. Padrah, Z.; Pastrav, A.; Palade, T.; Ratiu, O.; Puschita, E. Development and Validation of an ISA100. 11a Simulation Model for Accurate Industrial WSN Planning and Deployment. *Sensors* **2021**, *21*, 3600. [CrossRef] [PubMed]
16. Wang, R.; Guan, Y.; Li, X.; Zhang, R. Formal Verification of CAN Bus in Cyber Physical System. In Proceedings of the 2020 IEEE 20th International Conference on Software Quality, Macau, China, 11–14 December 2020.
17. Gokhale, S.; Dalvi, A.; Siddavatam, I. Industrial Control Systems Honeypot: A Formal Analysis of Conpot. *Int. J. Comput. Netw. Inf. Secur.* **2020**, *12*, 44–56. [CrossRef]
18. Szymoniak, S.; Kesar, S. Key Agreement and Authentication Protocols in the Internet of Things: A Survey. *Appl. Sci.* **2022**, *13*, 404. [CrossRef]
19. Jensen, K.; Kristensen, L.M.; Wells, L. Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems. *Int. J. Softw. Tools Technol. Transf.* **2007**, *9*, 213–254. [CrossRef]
20. Xu, Y.; Xie, X. Modeling and Analysis of Security Protocols Using Colored Petri Nets. *J. Comput.* **2011**, *6*, 19–27. [CrossRef]
21. Dudak, J.; Cicak, P. CPN model of the MODBUS protocol. In Proceedings of the 13th Mechatronika, Teplice, Slovakia, 2–4 June 2010.
22. Sadique, K.M.; Rahmani, R.; Johannesson, P. DIdM-EIoTD: Distributed Identity Management for Edge Internet of Things (IoT) Devices. *Sensors* **2023**, *23*, 4046. [CrossRef] [PubMed]
23. Gong, X.; Feng, T.; Du, J.Z. Formal modeling and security analysis method for security protocols based on CPN. *J. Commun.* **2021**, *42*, 240–253.
24. Gehlot, V.; Nigro, C. An introduction to systems modeling and simulation with colored petri nets. In Proceedings of the 2010 Winter Simulation Conference, Baltimore, MD, USA, 5–8 December 2010.
25. Al-Azzoni, I.; Down, D.G.; Khédri, R. Modeling and verification of cryptographic protocols using coloured petri nets and design/CPN. *Nord. J. Comput.* **2005**, *12*, 200–228.
26. Wu, D.; Liu, J.; Wang, H.; Tang, T. A cpn-based approach for studying impacts of communication delays on safety and availability of safety-critical distributed. *IEEE Trans. Ind. Inform.* **2021**, *18*, 3033–3042. [CrossRef]
27. Simonsen KI, F.; Kristensen, L.M. Towards a CPN-based modelling approach for reconciling verification and implementation of protocol models. In Proceedings of the Model-Based Methodologies for Pervasive and Embedded Software: 8th International Workshop, MOMPES 2012, Essen, Germany, 4 September 2012; Revised Papers 8. Springer: Berlin/Heidelberg, Germany, 2013; pp. 106–125.
28. Choppy, C.; Dedova, A.; Evangelista, S.; Klaï, K.; Petrucci, L.; Youcef, S. Modelling and formal verification of the NEO protocol. In *Transactions on Petri Nets and Other Models of Concurrency VI*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 197–225.
29. Sa, R.; Xilin, B.; Zhao, Y.; Menke, N. The LDP Protocol Formal Description and Verification Based on CPN Model. In Proceedings of the 4th International Conference on Computer Engineering and Networks: CENet2014, Shanghai, China, 19–20 July 2014; Springer International Publishing: Cham, Switzerland, 2015; pp. 305–314.
30. Ding, Y.; Su, G. A Reduction method for Verification of Security Protocol through CPN. In Proceedings of the 2008 IEEE International Conference on Networking, Sensing and Control, Sanya, China, 6–8 April 2008; IEEE: Piscataway, NJ, USA, 2008; pp. 73–77.

31. Carrasquel, J.C.; Morales, A.; Villapol, M.E. Prosega/CPN: An extension of CPN Tools for automata-based analysis and system verification. *Труды Института Системного Программирования Ран* **2018**, *30*, 107–128. [CrossRef] [PubMed]
32. Rodríguez, A.; Kristensen, L.M.; Rutle, A. Formal modelling and incremental verification of the MQTT IoT protocol. In *Transactions on Petri Nets and Other Models of Concurrency XIV*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 126–145.
33. Fan, Y.; Su, G.; Liu, H.; Zhu, S. Study on a CPN-based Auto-analysis Tool for Security Protocols. In Proceedings of the 2012 Fourth International Symposium on Information Science and Engineering, Shanghai, China, 14–16 December 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 179–182.
34. Kang, H.; Yang, X.; Yuan, S. Modeling and verification of web services composition based on cpn. In Proceedings of the 2007 IFIP International Conference on Network and Parallel Computing Workshops (NPC 2007), Dalian, China, 18–21 September 2007; IEEE: Piscataway, NJ, USA, 2007; pp. 613–617.