Formal Security Evaluation and Research of Automotive CAN Protocol Based on CPN

Tao Feng, Lu Zheng, and Peng-Shou Xie

(Corresponding author: Lu Zheng)

Faculty of Computer and Communication, Lanzhou University of Technology, China

Email: zhengl0518@163.com

(Received Aug. 31, 2021; Revised and Accepted Jan. 13, 2022; First Online Feb. 26, 2022)

Abstract

Controller Area Network (CAN) bus is the most representative in-vehicle bus technology in Intra-Vehicular Networks (IVNs) for its high reliability. However, the continuous increment of complex electronic systems makes the IVNs vulnerable to various malicious attacks. This work proposes a protocol model detection method based on a combination of Colored Petri Nets (CPN) and the Dolev-Yao attack model to a CAN2.0B-based IVN protocol for formal security evaluation. The results show that this protocol is vulnerable to two types of man-in-the-middle attacks: replay and spoofing. To address this, we use the asymmetric cryptosystem, and digital signature combined with the HASH function for reinforcement and again use the protocol model detection method to evaluate the security of the new protocol. The results show that the new protocol can effectively improve security.

Keywords: CAN Protocol; CPN Tools; Dolev-Yao; Formal Analysis; Security Evaluation

1 Introduction

With the fusion of vehicles and information technology, more systems and functions of ordinary vehicles are transformed from mechanical systems to electronic systems. Vehicle sensors, engine control and Anti-lock Brake System (ABS) and Advanced Driver-Assistance System (ADAS) [3], etc. have been put under the control of Electronic Control Units (ECUs) [7]. Manufacturers have transitioned from wire-heavy, point-to-point schemes to the bus, such as LIN, CAN, FlexRay, and MOST [22].

CAN is the international standard for communication inside the vehicle due to its high reliability, high fault tolerance, real-time, flexibility, etc. [9]. However, the security attributes of the CAN bus are mainly designed to ensure reliable communication. There are no security attributes such as encryption, authentication, integrity, and confidentiality. In recent years, the continuous improvement of functional requirements has caused a large increase in ECUs. [19] At the same time, the continuous increase of open interfaces makes IVN faces huge risks. Attackers can launch direct or indirect, long-distance or short-distance attacks on cars through OBD ports, WiFi, Bluetooth, cellular networks, etc. [18], and steal private data and cause serious harm. However, simply adding firewall protection to the vehicle gateway cannot fundamentally prevent malicious attacks. Therefore, the core focus of in-vehicle security is to protect the information security of the CAN protocol in the vehicle.

The rest of the paper is organized as follows. Section 2 discusses the security issues described in related work. Section 3 presents the CAN frame and security requirements. Section 4 models the CAN2.0B-based IVN protocol [17] and presents consistency verification based on CPN. Section 5 introduces the attacker model and describes the formal security analysis for the CAN2.0B-based IVN Protocol. Section 6 designs and models the new protocol. Section 7 presents the formal security analysis based on CPN. Section 8 compares our model detection method and the security attributes of our protocol with other work. Section 9 concludes the work.

2 Related Work

To build a safe in-vehicle CAN communication environment, in the past ten years, the European Union and other organizations have funded and conducted many projects to deal with in-vehicle network security, such as E-safety Vehicle Intrusion Protected Applications (EVITA) [8], Secure Vehicle Communication (SEVECOM), Open Vehicular Secure Platform (OVERSEE), etc. EVITA designs and verifies the architecture of the vehicle network, and develops a hardware security module (HSM) [8]. SEVECOM studies the protection of vehicle sensor data based on vehicle security middleware from the perspective of threat analysis and security architecture. OVERSEE provides the design of a vehicle application and communication platform, and builds a standardized in-vehicle environment [5]. However, this type of project did not study the specific security mechanism for the vehicle network, and did not propose the corresponding security protocol.

Security protocols in [10, 15, 23] were designed for the limited data load of the CAN data frames. However, these protocols do not support real-time data processing, and there is no corresponding formal analysis and security evaluation for these protocols. In [14], Radu and Garcia proposed a protocol to generate session keys, but the protocol lacked entity authentication and the specific formal analysis of the protocol. Woo *et al.* [17] proposed a set of CAN2.0B-based generic IVN protocols to solve the problem of providing a safe and fast key distribution mechanism for IVN. Basker *et al.* [13] used Tamarin Prover for formal security verification for this protocol, but it lacks intuitiveness.

To sum up, most of the existing research work on vehicle network security focuses on adding security mechanisms to achieve corresponding security functions, and the security evaluation and consistency verification of the protocol is still in its infancy. This paper focuses on the existing CAN2.0B-based IVN protocol [17], based on the theory of combining the colored Petri net (CPN) and the Dolev-Yao attack model in [1], conducts the security evaluation of the protocol and discovers its potential loopholes, fundamentally improves the protocol and prevents malicious attacks.

Compared with the existing research results, the main contributions of this paper include three aspects:

1) We propose a protocol model detection method that combines a colored Petri net (CPN) and the Dolev-Yao attack model;

2) We use the CPN Tools to perform CPN-based formal modeling of the CAN2.0B-based IVN protocol based on its specification, and introduce the Dolev-Yao attacker model to conduct security evaluation and perform consistency verification of the protocol model;

3) For the various vulnerabilities exposed after security evaluation of the protocol, we propose a new set of lightweight security enhancement protocols, and perform formal modeling of the new protocol. We verify functional and security by introducing the Dolev-Yao attacker model;

3 CAN Frames and CAN Security Requirements

3.1 CAN Frames

CAN bus was developed by German BOSCH company [4] in 1986, it has become the world's mainstream bus generally recognized by automobile manufacturers in various countries and has been set as an international standard. Figure 1 shows the CAN2.0B data frame format.

The CAN bus protocol supports two message formats, which can be divided into 11-bit CAN2.0A standard frame format and 29-bit CAN2.0B extended frame format according to the difference of ID fields [23]. The CAN2.0B protocol is compatible with the CAN2.0A, and is compatible with data messages in standard frame format and extended frame format at the same time. Since the data field defined by the CAN protocol can contain 0-8 bytes [11], the time consumed during data transmission is relatively short, thereby the real-time nature of data transmission is guaranteed. CAN protocol uses CRC to provide error checking in data transmission. However, only using CRC can only ensure the accuracy of the transmitted data, but cannot ensure reliable transmission [12].

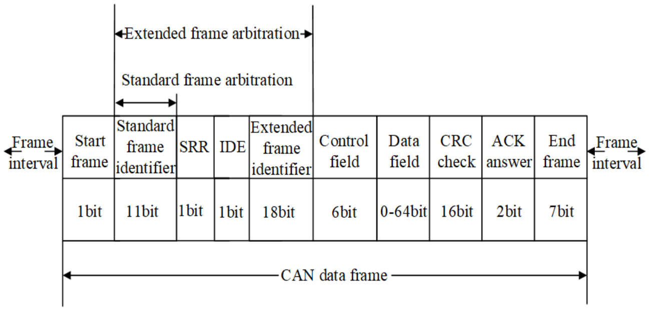

Figure 1: CAN2.0B data frame

3.2 CAN Security Requirements

A large number of researches on in-vehicle network security point out that the lack of data encryption and node authentication is the most serious vulnerability of CAN. However, due to the limitations of real-time and the limited data payload of CAN data frames and the limited memory in the ECU [11], achieving data authentication on CAN is a huge challenge [7]. In addition, malicious nodes can easily launch tampering attacks and replay attacks by stealing the ID of the bus node. The CRC on the CAN bus cannot ensure safe and accurate data transmission. Therefore, security protocols need to encrypt and authenticate data frames to prevent tampering attacks and replay attacks.

For the defects that CAN cannot encrypt data, cannot provide identity authentication for nodes, and cannot ensure the correctness and integrity of data, combined with the design characteristics of CAN bus, the security functions added on the original CAN protocol need to meet the following points of security demand:

1) Confidentiality. Each data frame in the CAN bus should be encrypted;

2) The authenticity of identity. The two parties who need to communicate in the CAN bus should confirm the identity of the other party before communication to prevent attackers from pretending to be a normal node to steal private data on the bus;

3) Correctness. The communication receiver in the CAN bus should ensure that the message comes from the correct sender;

4) Completeness. The communication receiver in the CAN bus should ensure that the message has not been tampered with by the attackers during the transmission;

5) Real-time. Due to the limited data payload of the CAN data frame, the freshness of the data transmitted on the CAN bus needs to be ensured to prevent replay attacks.

3.3 CPN Tools

The formal analysis method uses mathematics or logical structure to describe the system model, and verifies whether the system meets the requirements of consistency and completeness through a certain form of reasoning. Early methods used for formal analysis of protocols, such as BAN logic, string space, state machines, etc. The formal verification of these methods focuses on the theorem proving, without specific analysis of formal semantics. In recent years, powerful analysis tools [20] such as ProVerif, Scyther, Tamarin Prover, and CPN Tools [16] have emerged, which can perform formal security verification and semantic analysis for protocols.

This paper uses CPN Tools to edit, simulate, and analyze colored Petri nets, perform state space analysis and performance analysis, and model, analyze, and verify protocols. The programming language of CPN is based on the standard Markup Language (ML) [2] language, CPN Tools is similar to a state machine, and can simulate and analyze concurrent systems. Because of its powerful state space analysis [21] capabilities and easy-to-understand expressions, it has become one of the mainstream formal modeling and analysis tools for security protocols, and is widely used in many fields.

4 CAN2.0B-based IVN Protocol

The existing CAN2.0B-based IVN protocol [17] uses the security mechanism of HMAC combined with the KDF key derivation function. The communication process of the protocol is divided into three protocols, including the initial session key distribution phase (ISDP), the session key update phase (SKUP), and the data frame phase (DFP).

Inside the protocol, K_i and GK are symmetric keys shared by ECUi and GECU. EK_k is the encryption key for the kth session. AK_k is the authentication key for the kth session. KEK_k is the encryption key for the kth session in the key update phase. KGK_k is the generated key of the kth session in the key update phase. CTR_{GECU} is the data frame counter of GECU. M is the plaintext of the data frame. C is the ciphertext of the data frame. $KDF_x()$ is the key derivation function. $H_x()$ is the HASH function.

4.1 Modeling CAN2.0B-based IVN Protocol Based on CPN

When building large-scale models of protocol, using a traditional CPN single-page model is not only very complicated, but also not intuitive. To solve this problem, this paper adopts the idea of building hierarchical models, and replaces a module with substitution transitions of the CPN Tools in the high-level model, and builds multilayer system models with substitution transitions.

In this paper, the double-layer rectangles represent substitution transitions. The ovals represent the message places, which are used to store messages in the communication process. The arrows represent message transmission, which is used to transmit messages in the communication process.

We modeled the protocol into three levels: top, middle and bottom. The top-level CPN model of the protocol is shown in Figure 2. It intuitively simulates the process of the entire protocol.

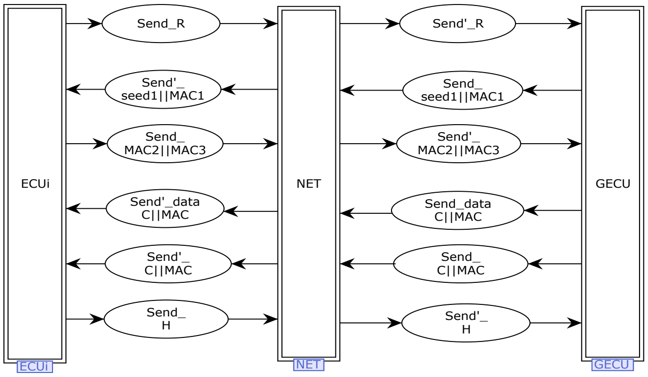

Figure 2: Top-level CPN model

The internal CPN model of the substitution transition NET is shown in Figure 3. The arrows represent the direction of data transmission.

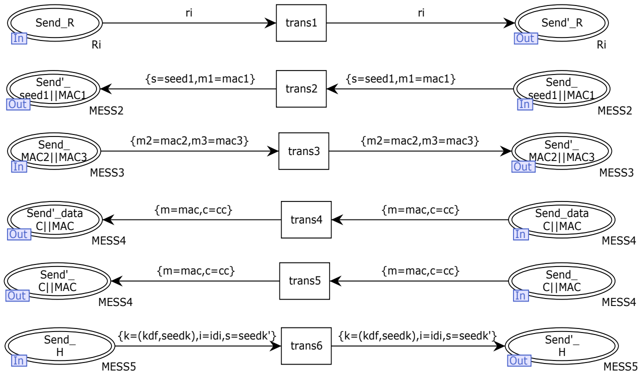

Figure 3: CPN model of substitution transition NET

The middle-level CPN model of the protocol is shown in Figure 4. It consists of 1 transition, 8 substitution transitions, and 14 places. Transition start means that ECUi selects the random number R_i and sends it to GECU. Substitution transitions Initialize and Initialize' represent the process of generating the initial session key. Substitution transitions date and date' represent the process of data frame transmission. Substitution transitions update and update' represent the process of updating the session key. Substitution transition cycle is responsible for storing and integrating the updated key seeds in the SKUP phase, and real-time updating of the keys used in the DFP phase.

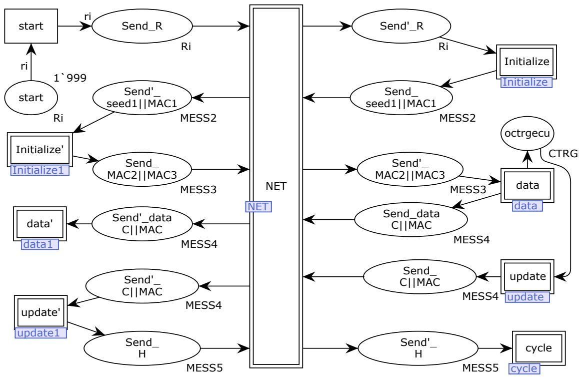

Figure 4: Mid-level CPN model

The bottom-level CPN model of the protocol consists of 8 parts. The ISDP phase, DFP phase, and SKUP phase are described in turn according to the protocol process. The internal CPN model of the substitution transition Initialize is shown in Figure 5. It simulates the process of GECU selecting the random number $Seed_1$, using K_i and HMAC algorithm to generate MAC_1 from ID_{GECU}, ID_i, R_i and $Seed_1$, and sending it to ECUi with $Seed_1$.

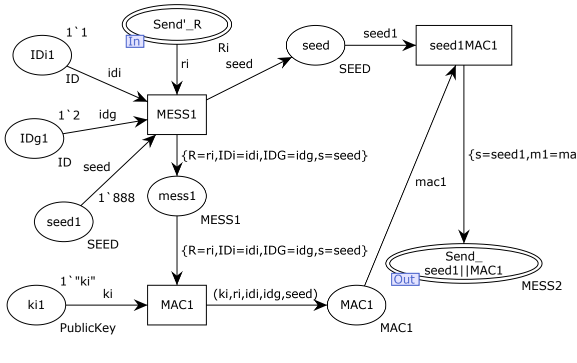

Figure 5: CPN model of substitution transition Initialize

The internal CPN model of the substitution transition Initialize' is shown in Figure 6. It simulates ECUi verifying MAC_1 and using $Seed_1$, GK and KDF to generate the initial session keys EK_1, AK_1, KEK_1 and KGK_1, and using K_i and HMAC algorithm to generate MAC_2 from ID_i and $Seed_1$, then using AK_1 and HMAC algorithm to generate MAC_3 from ID_i, EK_1, AK_1, KEK_1 and KGK_1, then sending MAC_2 and MAC_3 to GECU.

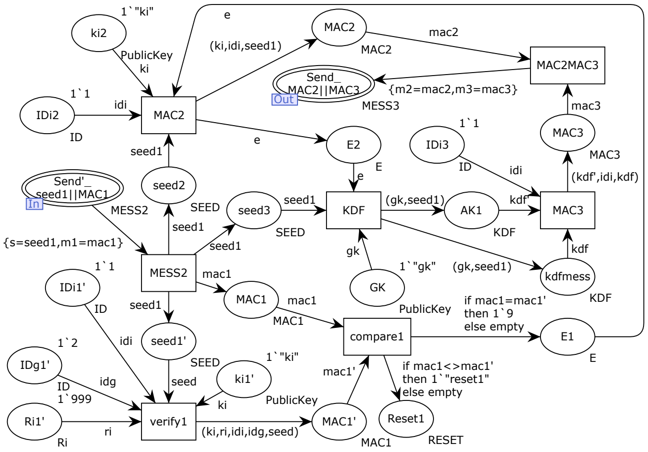

Figure 6: CPN model of substitution transition Initialize'

The internal CPN model of the substitution transition data is shown in Figure 7. It simulates GECU verifying MAC_2 and using KDF to calculate the initial session keys, and verifying MAC_3, and using EK_k and AES-128 to generate C from CTR_{GECU} and M, and using AK_k to generate MAC from ID_{GECU}, C and CTR_{GECU}, and sending it to ECUi with C, then increasing CTR_{GECU}.

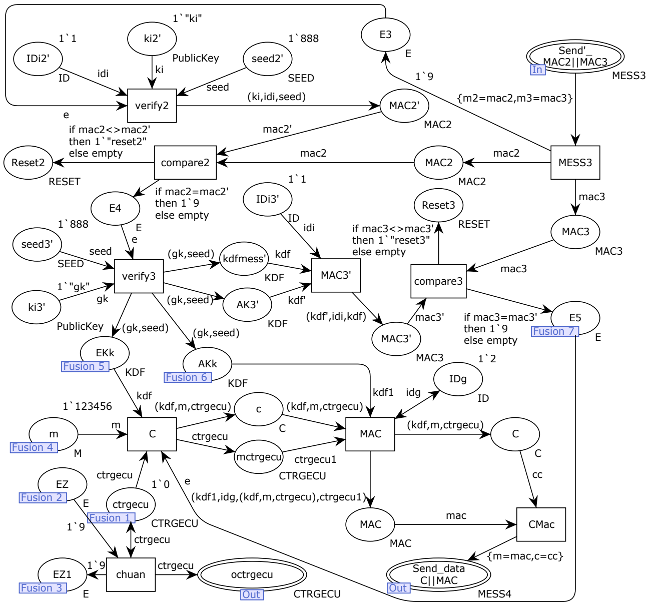

Figure 7: CPN model of substitution transition data

The internal CPN model of the substitution transition data' is shown in Figure 8. It simulates ECUi verifying MAC, decrypting to obtain M, increasing CTR_{GECU}.

The internal CPN model of the substitution transition update is shown in Figure 9. It simulates GECU selecting random number $Seed_{k+1}$, using KEK_k to generate C from CTR_{GECU} and $Seed_{k+1}$, using AK_k to generate MAC from ID_{GECU}, C, CTR_{GECU}, composing C and MAC into a key request message, sending it to ECUi.

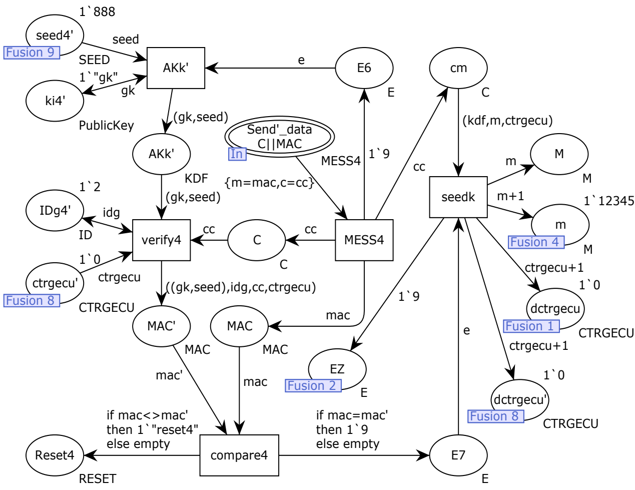

Figure 8: CPN model of substitution transition data'

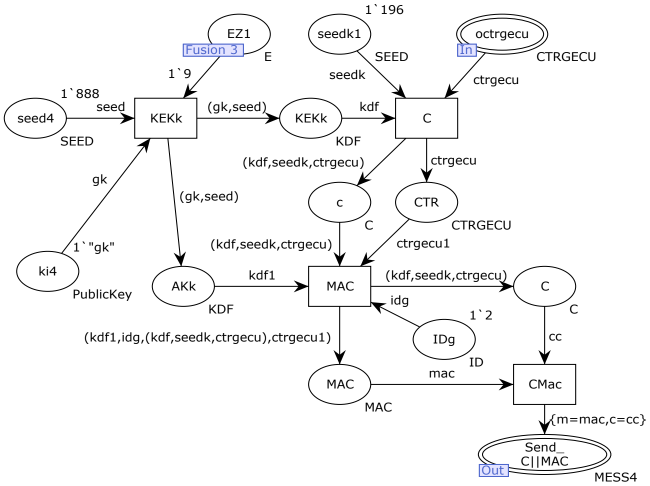

Figure 9: CPN model of substitution transition update

The internal CPN model of the substitution transition update' is shown in Figure 10. It simulates ECUi verifying MAC and decrypting C, and using KDF with KGK_k to derive the session key that will be used in the k+1 session, and initializing data frame counter to zero, then using AK_{k+1} and HMAC algorithm to generate a key response message from ID_i and $Seed_{k+1}$, and sending it to GECU.

4.2 Consistency Verification for the CPN Model

The consistency between the original CPN model and the protocol specification determines the accuracy of the security evaluation of the subsequent protocol model. We uses the state space analysis tool in CPN Tools to verify the consistency of the CPN model and the function of the protocol. The state space analysis report generated by the original CPN model is shown in Table 1.

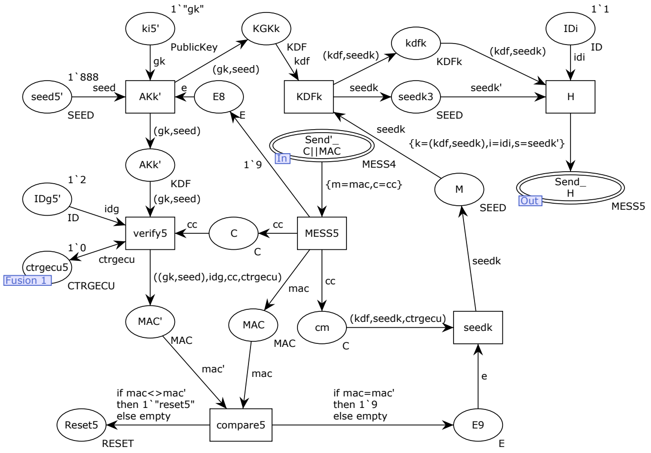

Figure 10: CPN model of substitution transition update'

Table 1: State Space report of the original model of CAN2.0B-based IVN protocol

Type	Number
State Space Node	56
State Space Arc	55
SCC Graph Node	56
SCC Graph Arc	55
MainState Space Node	0
Live Transition Instances	0
Dead Transition Instances	0
Dead Marking	1

In the state space report shown in Table 1, the number of nodes in the state space is the same as the number of strongly connected nodes, and the number of directed arcs in the state space is the same as the number of strongly connected arcs, it indicates that all state nodes of the original CPN model are reachable, and there is no state infinite loop behavior in the model. In the case of no attackers, the model will successfully perform initial key distribution, data frame transmission, trigger the key update and reply according to the protocol process. All interactive processes will not trigger the 5 reset places established in the model. The number of dead transitions is 0, which indicates that there is no situation where the transitions cannot be triggered. The number of dead markings is 1, which indicates that this model interacts following the protocol process, and there is 1 termination state, which is consistent with the expected result.

5 Security Evaluation of the Attacker Model

Dolev and Yao proposed an attacker model to verify the cryptographic protocols. So far, most of the attacker models introduced for the research of security protocols are based on the Dolev-Yao model. The Dolev-Yao attacker model points out that on the assumption that the cryptographic system is "absolutely secure", it does not study the security of the specific cryptographic algorithms of the protocol, but takes the inherent security properties of the protocol as the research goal.

The NET subpage in the original model of the protocol describes the data transmission path between ECUi and GECU in detail, it is equivalent to the network channel that the protocol information must pass through. Therefore, this paper introduces the Dolev-Yao attacker model through the NET subpage in the original model.

5.1 Introducing the Attacker Model

According to the powerful capabilities of the attacker in the Dolev-Yao attacker model, this paper launches three types of man-in-the-middle attacks, including replay, tampering, and spoofing on the network channel. The attacker model of the CPN model of the original protocol is shown in Figure 11.

The blue type of places and transitions represent the replay attack launched by Transition Attack1 and Attack2. The replay attack on the trans5 path is adopted an attacker model based on message splitting and combination, which can ensure the attacker's ability and effectively reduce the state space. The type of the place dis is DB, which stores the split and the unsplit message. Places P31, P32, P33, P34 store all kinds of atomic information. Transition TF uses the attacker's transition rules to store undecipherable messages in place P5. Transition TE uses the attacker's synthesis rules to synthesize and store atomic messages in place P5. The type of place P5 is CB, which stores information that cannot be decrypted and the information after the atomic message is synthesized. The red type of places and transitions represent the tampering attack launched by Transition Attack3. The place ASEED1 tampered with the key seed in the original message. The purple type of places and transitions represent the spoofing attack launched by all transitions trans1, trans2, trans3, trans4, trans5, and trans6 on the network transmission path.

5.2 Security Analysis of the Model

The state space reports of the original model and the attacker model are shown in Table 2. All state nodes of the attacker model are reachable, and there is no state infinite loop behavior in the model. The number of state space nodes and directed arcs of the attacker model has not increased significantly compared with the original model, it indicates that the introduction of the attacker model did not cause the state space to be too large to explode, it further shows that the introduced attacker model is effective.

The number of dead transitions of the attacker model is 0, which indicates that there is no unpredictable final state of the protocol process due to the man-in-the-middle attack. The number of dead markings is 10, it indicates that after the introduction of the attacker model, there are 10 termination states where the data transmission is completed.

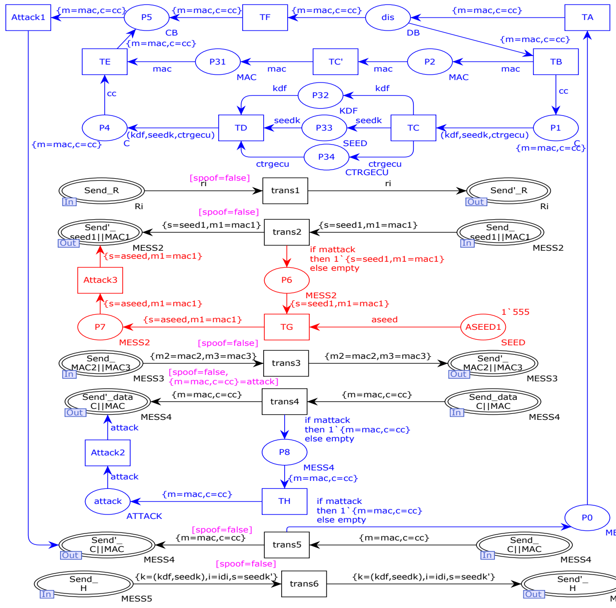

Figure 11: Formal description based on the attacker model

In the attacker model, the serial numbers of 10 dead markings can be obtained by writing the ML program and ListDeadMarking() function, all transitions and places corresponding to the serial number of each dead marking can be queried by NodeDescriptor() function. Through further analysis of the state of the places and transitions, we find that the 6 dead markings are caused by the replay attacks, including 3 dead markings are due to the lack of backward confidentiality caused by SKUP using the previous session key to update key, the remaining 3 dead markings are due to the lack of identity authentication caused by DFP only uses message verification to transmit data frames. The 4 dead markings are due to the attacker pretending to be a legal identity for message authentication caused by the spoofing attacks.

6 The New Protocol

6.1 The Message Flow Model of the New Protocol

In our new protocol, we introduce the asymmetric cryptosystem and digital signature for authentication to fix the vulnerability of the original protocol. we ensure the backward confidentiality of the session key. We use the HASH algorithm instead of HMAC to reduce the costs of computing and storage. The message flow model of the new protocol is shown in Figure 12.

Inside the model, PU_a and PR_a are the public and

Table 2: State Space reports of the original model and the attack model of CAN2.0B-Based IVN protocol

Type	Original Model	Original Attack Model
State Space Node	56	1288
State Space Arc	55	3343
SCC Graph Node	56	1288
SCC Graph Arc	55	3343
MainState Space Node	0	0
Live Transition Instances	0	0
Dead Transition Instances	0	0
Dead Marking	1	10

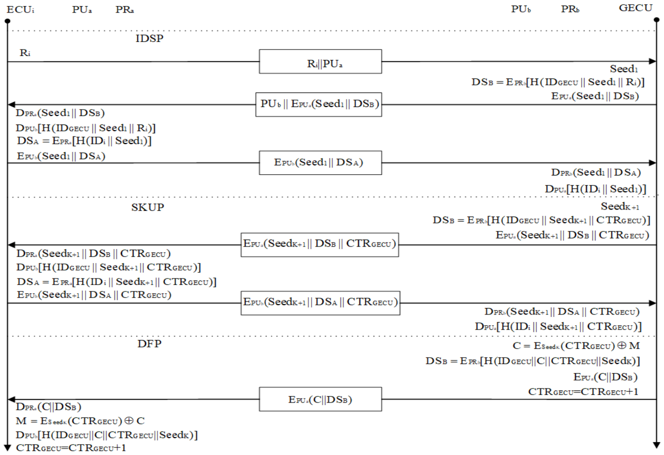
ECU_i
PU_a
PR_a
PU_b
PR_b
GECU
IDSP
R_i
$R_i||PU_a$
$Seed_1$
$DS_B = E_{PR_b}[H(ID_{GECU} || Seed_1 || R_i)]$
$E_{PU_a}(Seed_1 || DS_B)$
$PU_b || E_{PU_a}(Seed_1 || DS_B)$
$D_{PR_a}(Seed_1 || DS_B)$
$D_{PU_b}[H(ID_{GECU} || Seed_1 || R_i)]$
$DS_A = E_{PR_a}[H(ID_i || Seed_1)]$
$E_{PU_b}(Seed_1 || DS_A)$
$E_{PU_b}(Seed_1 || DS_A)$
$D_{PR_b}(Seed_1 || DS_A)$
$D_{PU_a}[H(ID_i || Seed_1)]$
SKUP
$Seed_{K+1}$
$DS_B = E_{PR_b}[H(ID_{GECU} || Seed_{K+1} || CTR_{GECU})]$
$E_{PU_a}(Seed_{K+1} || DS_B || CTR_{GECU})$
$E_{PU_a}(Seed_{K+1} || DS_B || CTR_{GECU})$
$D_{PR_a}(Seed_{K+1} || DS_B || CTR_{GECU})$
$D_{PU_b}[H(ID_{GECU} || Seed_{K+1} || CTR_{GECU})]$
$DS_A = E_{PR_a}[H(ID_i || Seed_{K+1} || CTR_{GECU})]$
$E_{PU_b}(Seed_{K+1} || DS_A || CTR_{GECU})$
$E_{PU_b}(Seed_{K+1} || DS_A || CTR_{GECU})$
$D_{PR_b}(Seed_{K+1} || DS_A || CTR_{GECU})$
$D_{PU_a}[H(ID_i || Seed_{K+1} || CTR_{GECU})]$
DFP
$C = E_{Seed_K}(CTR_{GECU}) \oplus M$
$DS_B = E_{PR_b}[H(ID_{GECU}||C||CTR_{GECU}||Seed_K)]$
$E_{PU_a}(C||DS_B)$
$CTR_{GECU}=CTR_{GECU}+1$
$E_{PU_a}(C||DS_B)$
$D_{PR_a}(C||DS_B)$
$M = E_{Seed_K}(CTR_{GECU}) \oplus C$
$D_{PU_b}[H(ID_{GECU}||C||CTR_{GECU}||Seed_K)]$
$CTR_{GECU}=CTR_{GECU}+1$

Figure 12: Message flow model of the new protocol

private keys generated by ECUi respectively, PU_b and PR_b are the public and private keys generated by GECU respectively, the pair of public and private keys are distributed with the aid of the KPI architecture. DSB is the digital signature generated by GECU. DSA is the digital signature generated by ECUi. M is CAN data frame. CTR_{GECU} is the data frame counter of GECU. $H_x()$ is the HASH function. The process of ISDP is shown as follows:

1) ECUi selects a random number R_i and sends it to GECU together with PU_a;

2) GECU selects a random number $Seed_1$, generates the $hash1$ from ID_GECU and $Seed_1$, uses PR_b to generate the DSB from $hash1$, and uses PU_a to encrypt Seed1 and DSB, and transmits them to ECUi together with the PU_b;

3) ECUi uses PR_a to decrypt to obtain $Seed_1$ and DSB, uses PU_b to decrypt to obtain the $hash1$. ECUi verifies $hash1$;

4) ECUi generates the $hash2$ from ID_i and $Seed_1$, uses PR_a to generate the DSA from $hash2$, uses PU_b to encrypt $Seed_1$ and DSA, and transmits them to GECU;

5) GECU uses PR_b to decrypt to obtain $Seed_1$ and DSA, uses PU_a to decrypt to obtain $hash2$, GECU verifies $hash2$;

The process of SKUP is shown as follows:

1) GECU selects a random number $Seed_{k+1}$, generates $hash3$ from ID_{GECU}, $Seed_{k+1}$ and CTR_{GECU}, uses PR_b to generate the DSB from $hash3$, and uses PU_a to encrypt $Seed_{k+1}$, CTR_{GECU} and DSB, and sends them to ECUi;

2) ECUi uses PR_a to decrypt to obtain $Seed_{k+1}$, DSB, CTR_{GECU}, uses PU_b to decrypt to obtain $hash3$. ECUi verifies $hash3$;

3) ECUi generates $hash4$ from ID_i, $Seed_{k+1}$ and CTR_{GECU}, uses PR_a to generate DSA from $hash4$, uses PU_b to encrypt $Seed_{k+1}$, CTR_{GECU} and DSA, and sends them to GECU;

4) GECU uses PR_b to decrypt to obtain $Seed_{k+1}$, CTR_{GECU} and DSA, uses PU_a to decrypt to obtain $hash4$. GECU verifies $hash4$;

The process of DFP is shown as follows:

1) GECU uses Seedk as the key to transmits M, uses the AES-128 to generate C from CTRGECU and M based on

$$C = E_{Seed_k}(CTR_{GECU}) \oplus M$$

GECU generates the $hash5$ from ID_GECU, C, $Seed_k$ and CTR_{GECU}, uses PR_b to generate the DSB from $hash5$, uses PU_a to encrypt C and DSB, and sends them to ECUi. GECU increases the CAN data frame counter (CTR_{GECU});

2) ECUi uses PR_a to decrypt to obtain C and DSB, decrypts to obtain the plaintext M based on

$$M = E_{Seed_k}(CTR_{GECU}) \oplus C$$

ECUi uses PU_b to decrypt to obtain the $hash5$, ECUi verifies $hash5$. ECUi increases CTR_{GECU};

6.2 Modeling the New Protocol Based on CPN

We model the new protocol based on CPN to verify whether its safety mechanism meets the safety requirements. The middle-level CPN model of the new protocol is shown in Figure 13. Transition start represents the process of ECUi and GECU generating public and private keys respectively. Substitution transitions Initialize and Initialize' represent the process of completing the initial session key distribution and identity authentication. Substitution transition date and date' represent the process of completing the secure transmission of the CAN data frames. Substitution transition update and update' represent the process of completing the session key update.

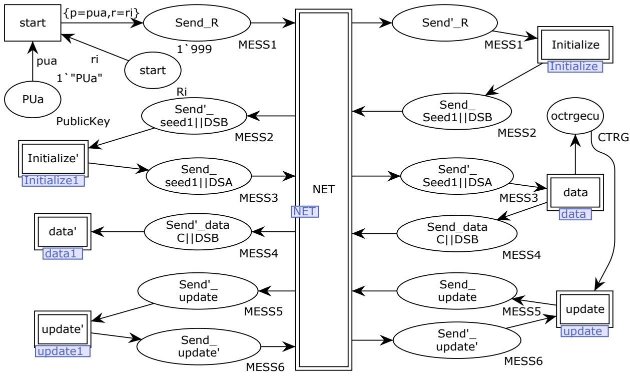

Figure 13: Middle-level CPN model of the new protocol

The internal CPN model of the substitution transition Initialize is shown in Figure 14. Transition HASH1 combines ID_{GECU}, R_i, and $Seed_1$ into $hash1$. Transition DSB1 combines $hash1$ and PR_b into DSB_1. Transition seed1DSB1 combines PU_a, $Seed_1$, DSB_1, and PU_b, and sends them to ECUi through the place Send_Seed1‖DSB.

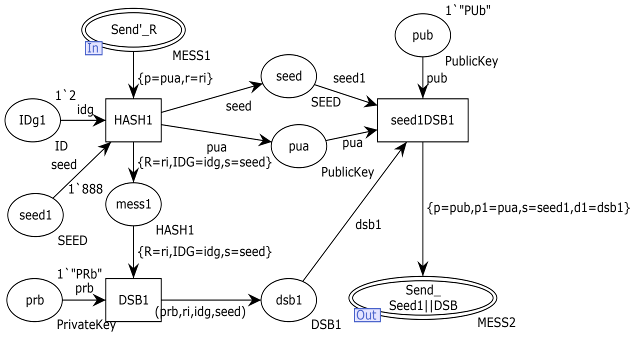

Figure 14: CPN model of new protocol's substitution transition Initialize

The internal CPN model of the substitution transition Initialize' is shown in Figure 15. Transition compare1 verifies hash1 in the DSB_1. If the verification succeeds, transition HASH2 combines $Seed_1$ and ID_i into $hash2$, else place Reset1 is triggered. Transition DSA1 combines $hash2$ and PR_a into DSA_1. Transition seed1DSA combines PU_b, $Seed_1$, and DSA_1, and sends them to GECU through the place Send_seed1‖DSA.

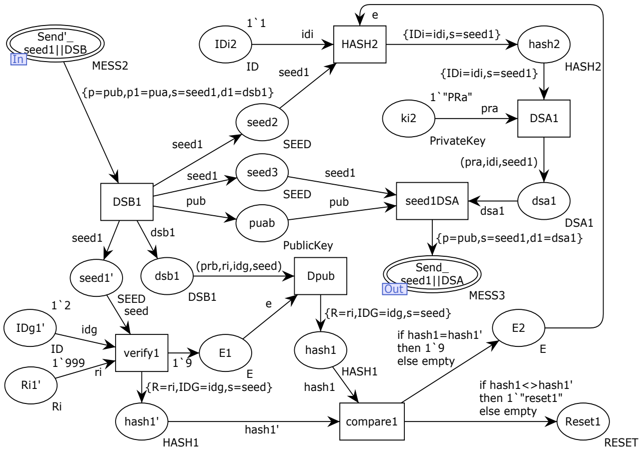

Figure 15: CPN model of new protocol's substitution transition Initialize'

The internal CPN model of the substitution transition data is shown in Figure 16. Transition compare2 verifies $hash2$ in the DSA_1. If the verification succeeds, transition C combines m, $ctrgecu$ and $Seed_k$ into C, else place Reset2 is triggered. Transition HASH3 combines ID_{GECU}, C, $ctrgecu$, and $Seed_k$ into $hash3$. Transition DSB2 combines $hash3$ and PR_b into DSB_2. Transition CDSB combines PU_a, C, and DSB_2, and sends them to ECUi through the place Send_dataC‖DSB. Transition chuan is responsible for updating the data frame counter in time when a data frame is transmitted.

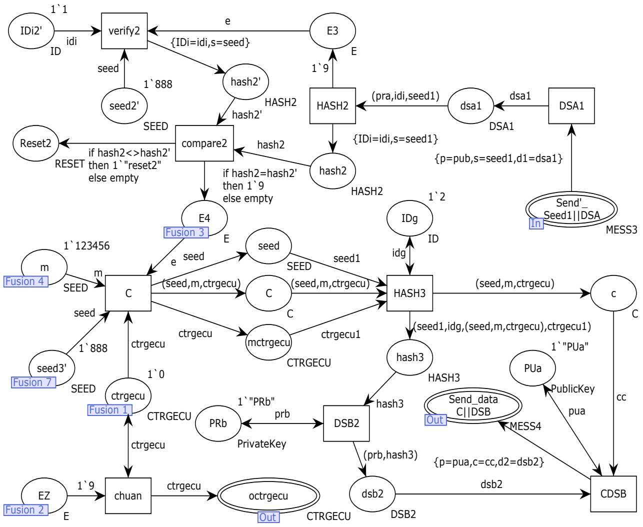

Figure 16: CPN model of new protocol's substitution transition data

The internal CPN model of the substitution transition data' is shown in Figure 17. Transition compare3 verifies $hash3$ in the DSB_2. If the verification succeeds, the transition M decrypts C so that ECUi obtains M, else place Reset3 is triggered. At this time, a data frame has been transmitted, the data frame counter stored in the place is increased by 1.

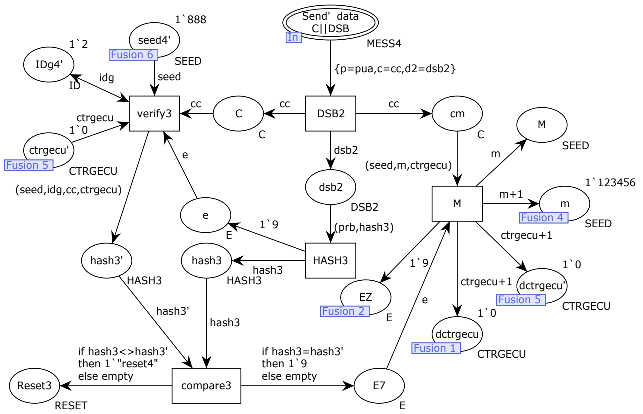

Figure 17: CPN model of new protocol's substitution transition data'

The internal CPN model of the substitution transition update is shown in Figure 18. Transition HASH4 combines ID_{GECU}, $ctrgecu$, and $Seed_k$ into $hash4$. Transition DSB3 combines $hash4$ and PR_b into DSB_3. Transition update combines PU_a, $Seed_k$, and DSB_3, and sends them to ECUi through the place Send_update. Transition DSA2 verifies $hash5$ in the DSA_2. If the verification succeeds, the SKUP is completed and place E4 reaches the subpage of substitution transition data to trigger the next data frame transmission, else place Reset5 is triggered.

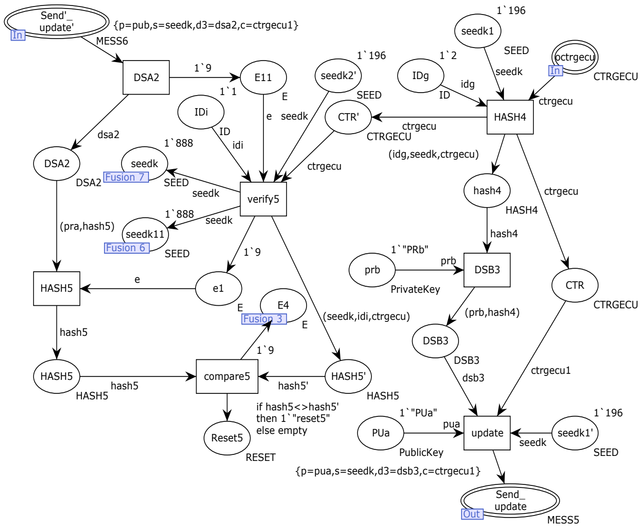

Figure 18: CPN model of new protocol's substitution transition update

The internal CPN model of the substitution transition update' is shown in Figure 19. Transition compare4 verifies $hash4$ in the DSB_3. If the verification succeeds, transition HASH5 combines ID_i, $ctrgecu$, and $Seed_k$ into $hash5$, else place Reset4 is triggered. Transition DSA2 combines $hash5$ and PR_a into DSA_2. Transition H combines PU_b, $ctrgecu$, $Seed_k$, and DSA_2, and sends them to GECU through the place Send_update'.

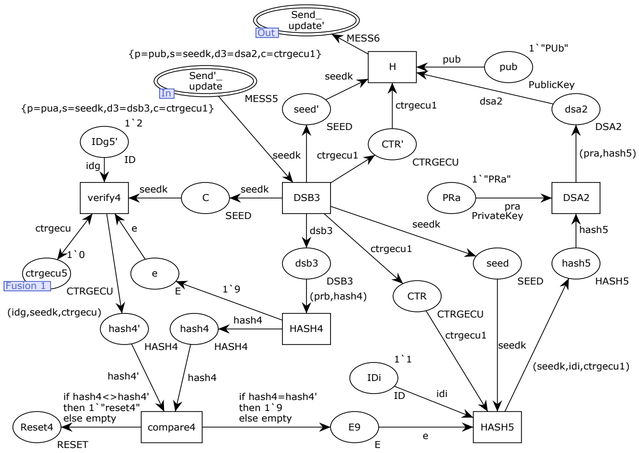

Figure 19: CPN model of new protocol's substitution transition update'

7 Formal Security Evaluation of the New Protocol

7.1 Introducing the Attacker Model

In the new protocol, the same method is used to introduce the Dolev-Yao attacker model. We launch man-in-the-middle attacks of replay, tampering, and spoofing to the network channel. The attacker model of the CPN model of the new protocol is shown in Figure 20. Inside the model, the blue type of places and transitions represent the replay attack, the red type of places and transitions represent the tampering attack, the purple type of places and transitions represent the spoofing attack.

7.2 Security Analysis of the New Protocol

The state space reports of the attacker model of the new protocol and the attacker model of the original protocol are shown in Table 3. There was no explosion of the state space indicates that the attacker model was effectively introduced. The number of dead markings in the attacker model of the new protocol is 1, which indicates that there is only 1 termination state after the data transmission is completed. It further shows that the new protocol has no other attack status and can resist man-in-the-middle attacks of replay, tampering, and spoofing.

Table 3: State Space reports of the attacker model of original protocol and new protocol

Type	Original Attack Model	New Attack Model
State Space Node	1288	2035
State Space Arc	3343	4974
SCC Graph Node	1288	2035
SCC Graph Arc	3343	4974
MainState Space Node	0	0
Live Transition Instances	0	0
Dead Transition Instances	0	0
Dead Marking	10	1

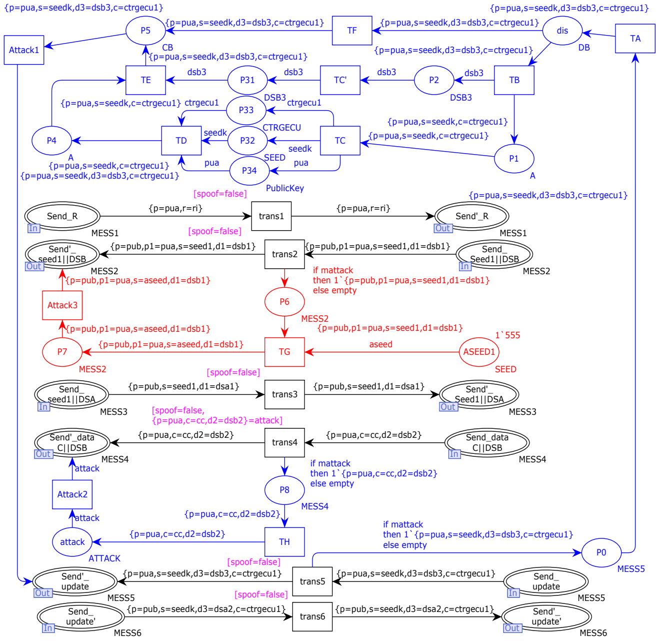
{p=pua,s=seedk,d3=dsb3,c=ctrgecu1}
Attack1
P5
{p=pua,s=seedk,d3=dsb3,c=ctrgecu1}
TF
{p=pua,s=seedk,d3=dsb3,c=ctrgecu1}
dis
{p=pua,s=seedk,d3=dsb3,c=ctrgecu1}
TA
DB
CB
{p=pua,s=seedk,d3=dsb3,c=ctrgecu1}
{p=pua,s=seedk,d3=dsb3,c=ctrgecu1}
TE
dsb3
P31
dsb3
TC'
dsb3
P2
dsb3
TB
DSB3
{p=pua,s=seedk,c=ctrgecu1}
ctrgecu1
P33
DSB3
ctrgecu1
P4
TD
seedk
P32
CTRGECU
TC
{p=pua,s=seedk,c=ctrgecu1}
{p=pua,s=seedk,c=ctrgecu1}
seedk
P1
A
A
{p=pua,s=seedk,c=ctrgecu1}
{p=pua,s=seedk,d3=dsb3,c=ctrgecu1}
pua
P34
SEED
pua
PublicKey
[spoof=false]
{p=pua,s=seedk,d3=dsb3,c=ctrgecu1}
Send_R
In
MESS1
{p=pua,r=ri}
trans1
{p=pua,r=ri}
Send'_R
Out
MESS1
[spoof=false]
Send'_ seed1||DSB
Out
MESS2
{p=pub,p1=pua,s=seed1,d1=dsb1}
trans2
{p=pub,p1=pua,s=seed1,d1=dsb1}
Send_ Seed1||DSB
In
MESS2
{p=pub,p1=pua,s=aseed,d1=dsb1}
if mattack
then 1`{p=pub,p1=pua,s=seed1,d1=dsb1}
else empty
Attack3
P6
MESS2
{p=pub,p1=pua,s=aseed,d1=dsb1}
{p=pub,p1=pua,s=seed1,d1=dsb1}
1`555
P7
{p=pub,p1=pua,s=aseed,d1=dsb1}
TG
aseed
ASEED1
MESS2
SEED
[spoof=false]
Send_ seed1||DSA
In
MESS3
{p=pub,s=seed1,d1=dsa1}
trans3
{p=pub,s=seed1,d1=dsa1}
Send'_ Seed1||DSA
Out
MESS3
[spoof=false,
{p=pua,c=cc,d2=dsb2}=attack]
Send'_data C||DSB
Out
MESS4
{p=pua,c=cc,d2=dsb2}
trans4
{p=pua,c=cc,d2=dsb2}
Send_data C||DSB
In
MESS4
attack
if mattack
then 1`{p=pua,c=cc,d2=dsb2}
else empty
Attack2
P8
MESS4
attack
{p=pua,c=cc,d2=dsb2}
attack
{p=pua,c=cc,d2=dsb2}
TH
if mattack
then 1`{p=pua,s=seedk,d3=dsb3,c=ctrgecu1}
else empty
P0
ATTACK
MESS5
[spoof=false]
Send'_ update
Out
MESS5
{p=pua,s=seedk,d3=dsb3,c=ctrgecu1}
trans5
{p=pua,s=seedk,d3=dsb3,c=ctrgecu1}
Send_ update
In
MESS5
[spoof=false]
Send_ update'
In
MESS6
{p=pub,s=seedk,d3=dsa2,c=ctrgecu1}
trans6
{p=pub,s=seedk,d3=dsa2,c=ctrgecu1}
Send'_ update'
Out
MESS6

Figure 20: Formal description of new protocol based on the attacker model

By writing the ML program to further analyze, it is found that the session key of the new protocol ensures freshness and backward confidentiality, so it can resist replay attacks. The new protocol can resist spoofing attacks by authentication of digital signatures. The new protocol can resist tampering attacks by the HASH function to verify the correctness of the message during data transmission.

8 Comparison of Analysis Methods and Performance Analysis of the New Protocol

We compare our proposed protocol with other CAN2.0B-based IVN protocols with high security. The comparison of security attributes is shown in Table 4.

Our scheme introduces the asymmetric cryptosystem, improves the defect that the original protocol completely relies on the secure channel to distribute symmetric keys, eliminates the potential risks brought by traditional symmetric cryptography, and reduces the difficulty of key distribution and the complexity of key management. The new protocol cancels the HMAC and uses the HASH algorithm for message verification, which reduces the cost of computing while also ensuring security. For the interactive process, the state space of the new protocol increased but did not explode. It indicates that the new protocol increases computing time within an acceptable range to improve security.

To verify the effectiveness of the formal analysis in our proposed model detection method used in this paper, we compare it with the other effective model detection methods of IVN protocols. The results are shown in Table 5.

Woo *et al.* can only achieve verification of the functional correctness of the protocol. Basker *et al.* [13] use the Tamarin tool to formally analyze the CAN2.0B-based IVN protocol, which can achieve anomaly detection and verification of the functional correctness of the protocol. Joe *et al.* [6] verify the proposed IVN protocol through the AVISPA tool, which can verify the functional correctness of the protocol and has intuitiveness. Our proposed protocol model detection method can achieve anomaly detection and show the attack types of the man-in-the-middle attacks. The intuitive graphics can accurately describe the steps of the protocol. The state space generated by formal analysis is intuitive and can detect the security attributes of the protocol and verify the functional correctness of the protocol. This model detection method can be used for security analysis and research of other IVN protocols.

9 Conclusion

This paper proposes a model detection method based on a combination of CPN and Dolev-Yao attack models for formally security evaluation of the Woo *et al.* CAN2.0B-based IVN protocol, and shows that it cannot resist the two types of man-in-the-middle attacks: replay and spoofing. To solve them, we propose a new protocol to ensure

Table 4: Comparison of protocol security attributes

Security attributes	**Ref [17]**	**Ref [13]**	**Ref [6]**	**Ref [14]**	**Ref [11]**	**Our Scheme**
Session key security	No	Yes	Yes	Yes	Yes	Yes
Entity authentication	No	No	No	No	Yes	Yes
Resistance to reply attack	No	Yes	No	No	No	Yes
Resistance to tampering attack	Yes	Yes	Yes	Yes	Yes	Yes
Resistance to spoofing attack	No	No	No	No	Yes	Yes
Provable security	No	Yes	Yes	Yes	No	Yes
Formal verification	No	Yes	No	No	Yes	Yes

Table 5: Comparison of protocol analysis methods

Scheme	**Anomaly detection**	**Attack type**	**Intuitive graphic**	**Verify functional correctness**	**State space**
Ref [17]	No	No	No	Yes	No
Ref [13]	Yes	No	No	Yes	No
Ref [6]	No	No	Yes	Yes	No
Ref [14]	Yes	No	No	Yes	No
Ref [11]	Yes	No	No	Yes	No
Our Scheme	Yes	Yes	Yes	Yes	Yes

session key security attributes: backward confidentiality and entity authentication. Finally, we give a comparison of our protocol with Woo *et al.* protocol by the state space reports of CPN Tools and writing ML language. The results show that our protocol does not affect the computing time and storage cost, and provides robust security than Woo *et al.* protocol, and can resist the three types of man-in-the-middle attacks: replay, spoofing and tampering.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 61762060).

References

[1] R. Amoah, S. Suriadi, S. Camtepe, and E. Foo, "Security analysis of the non-aggressive challenge response of the dnp3 protocol using a cpn model," in *ICC 2014 - 2014 IEEE International Conference on Communications*, 2014.

[2] I. Artamonov and A. Sukhodolov, "Cpn tools-based software solution for reliability analysis of processes in microservice environments," *International Journal of Simulation: Systems*, vol. 19, no. 6, 2019.

[3] L. L. Bello, R. Mariani, S. Mubeen, and S. Saponara, "Recent advances and trends in on-board embedded and networked automotive systems," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, pp. 1038–1051, 2019.

[4] M. Bozdal, M. Samie, S. Aslam, and I. Jennions, "Evaluation of can bus security challenges," *Sensors*, vol. 20, no. 2364, pp. 16–17, 2020.

[5] B. Groza and S. Murvay, "Efficient protocols for secure broadcast in controller area networks," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 2034–2042, 2013.

[6] J. Halabi and H. Artail, "A lightweight synchronous cryptographic hash chain solution to securing the vehicle can bus," in *IEEE International Multidisciplinary Conference on Engineering Technology*, 2018.

[7] M. Inam, Z. Li, A. Ali, and A. Zahoor, "A novel protocol for vehicle cluster formation and vehicle head selection in vehicular ad-hoc networks," *International Journal of Electronics and Information Engineering*, vol. 10, no. 2, pp. 103–119, 2019.

[8] R. Islam and R. Refat, "Improving can bus security by assigning dynamic arbitration ids," *Journal of Transportation Security*, vol. 13, 2020.

[9] S. Jadhav and D. Kshirsagar, "A survey on security in automotive networks," in *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, 2018.

[10] J. Lu, X. He, Y. Yang, D. Wang, and B. Meng, "Automatic verification of security of identity federation security protocol based on saml2.0 with proverif in the symbolic model," *International Journal of Network Security*, vol. 22, no. 1, pp. 80–92, 2020.

[11] S. Murvay and B. Groza, "Security shortcomings and countermeasures for the sae j1939 commercial vehi-

cle bus protocol," *IEEE Transactions on Vehicular Technology*, pp. 1–1, 2018.

[12] V. Odelu, A. K. Das, and A. Goswami, "A secure biometrics-based multi-server authentication protocol using smart cards," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 9, pp. 1–1, 2015.

[13] B. Palaniswamy, S. Camtepe, E. Foo, and J. Pieprzyk, "An efficient authentication scheme for intra-vehicular controller area network," *IEEE Transactions on Information Forensics and Security*, vol. PP, no. 99, pp. 1–1, 2020.

[14] A. I. Radu and F. D. Garcia, "Leia: A lightweight authentication protocol for can," *Springer International Publishing*, 2016.

[15] Ivan Edmar Carvajal Roca, Jian Wang, Jun Du, and Shuangqing Wei, "A semi-centralized security framework for in-vehicle networks," in *2020 International Wireless Communications and Mobile Computing (IWCMC)*, 2020.

[16] M. Simon, D. Moldt, D. Schmitz, and M. Haustermann, "Tools for curry-coloured petri nets," *International Conference on Applications & Theory of Petri Nets & Concurrency*, 2019.

[17] S. Woo, H. J. Jo, and H. L. Dong, "A practical wireless attack on the connected car and security protocol for in-vehicle can," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 1–14, 2014.

[18] S. Woo, H. J. Jo, I. S. Kim, and D. H. Lee, "A practical security architecture for in-vehicle can-fd," *IEEE Transactions on Intelligent Transportation Systems*, pp. 2248–2261, 2016.

[19] L. Xiao, X Lu, T. Xu, W. Zhuang, and H. Dai, "Reinforcement learning-based physical-layer authentication for controller area networks," *IEEE Transactions on Information Forensics and Security*, vol. PP, no. 99, pp. 1–1, 2021.

[20] L. Yao, J. Liu, D. Wang, J. Li, and B. Meng, "Formal analysis of sdn authentication protocol with mechanized protocol verifier in the symbolic model," *International Journal of Network Security*, vol. 20, no. 6, pp. 1125–1136, 2018.

[21] A Yg, A Jl, A Xl, A Yc, and J. B. Yan, "State space model identification of multirate processes with time-delay using the expectation maximization," *Journal of the Franklin Institute*, vol. 356, no. 3, pp. 1623–1639, 2019.

[22] C. Young, J. Zambreno, H. Olufowobi, and G. Bloom, "Survey of automotive controller area network intrusion detection systems," *IEEE Design and Test*, pp. 1–1, 2019.

[23] H. Zhang, X. Meng, X. Zhang, and Z. Liu, "Cansec: A practical in-vehicle controller area network security evaluation tool," *Sensors*, vol. 20, no. 17, p. 4900, 2020.

Biography

Tao Feng received the D.E from Xidian Univerisity, China, in 2008. He is a member of the China Computer Federation and China Cryptography Federation. He is currently a professor and doctoral supervisor at the Faculty of Computer and Communication, Lanzhou University of Technology, China. His research interests are network and information security, industrial internet with focus on technologies for provable theory of security protocols, design and implementation of security middleware and cyberspace application system, privacy protection.

Lu Zheng received the B.E. degrees from Hefei University of Technology, China, in 2019. She is currently enrolled as M.E. student at the Faculty of Computer and Communication in the same institution. Her research interests are information security and provable theory of security protocols.

Peng-shou Xie is currently a professor and master supervisor at the Faculty of Computer and Communication, Lanzhou University of Technology, China. His research interest is security on internet of vehicles and industrial internet.