Hindawi
Security and Communication Networks
Volume 2021, Article ID 2314520, 13 pages
https://doi.org/10.1155/2021/2314520

Research Article

A Scalable Security Protocol for Intravehicular Controller Area Network

Zi-An Zhao, Yu Sun, Dawei Li, Jian Cui, Zhenyu Guan, and Jianwei Liu

School of Cyber Science and Technology, Beihang University, Beijing 100191, China

Correspondence should be addressed to Yu Sun; sunyv@buaa.edu.cn

Received 16 September 2021; Revised 2 November 2021; Accepted 30 November 2021; Published 31 December 2021

Academic Editor: Qi Jiang

Intravehicular communication relies on controller area network (CAN) protocol to deliver messages and instructions among different electronic control units (ECU). Unfortunately, inherent defects in CAN include the absence of confidentiality and integrity mechanism, enabling adversaries to launch attacks from wired or wireless interfaces. Although various CAN cryptographic protocols have been proposed for entity authentication and secure communication, the redundancy in the key establishment phase weakens their availability in large-scale CAN. In this paper, we propose a scalable security protocol suite for intravehicular networks and reduce the communication costs significantly. A new type of attack, suspension attack, is identified for the existing protocols and mitigated in our protocol by leveraging a global counter scheme. We formally verify the security properties of the proposed protocol suite through the AVISPA tool. The simulation results indicate that the communication and computation efficiency are improved in our protocol.

1. Introduction

In a modern vehicle, hundreds of electronic control units (ECUs) are connected through in-vehicle network (IVN) gateway [1] to ensure life-critical operations, such as collision prediction and antilock braking. These ECUs communicate through controller area network (CAN) bus, the de facto standard for intravehicular communication, to guarantee driving security through sensing, actuation, and control. With the increasing number of sensor nodes, the function of the automobile is becoming much more complex, rendering more ECUs to be connected to the bus. In a luxury vehicle, more than 100 ECUs have been installed [2].

Inevitably, with more complex service and sophisticated communication functions incorporated into the automobile, the attack surface is growing rapidly. During the past decades, numerous researches demonstrated the ability to maliciously control a vehicle in road tests, from physical access to remote attack. As the most used automotive interface, onboard diagnostics (OBD) II provides direct and standard access to internal networks, through which CAN buses are accessible and physically exposed to an adversary [3, 4]. In addition, remote exploitation is feasible via multifarious attack vectors [5], including mechanics tools, Bluetooth, cellular radio, and internet connectivity such as Wi-Fi and 4G [6–8]. These interfaces, which frequently interact with the external world, provide potential entry points for an attacker to insert malicious messages and codes in CAN. Numerous cases [9–11] have been reported to attack the engine, brake, lamp, and fuel gauge on Ford, Toyota, and Tesla automobiles in both parking and driving scenarios to realize steering, braking, and acceleration and display control. Even the firmware and built-in code can be modified by the attackers.

The major security vulnerabilities for CAN bus are the lack of confidentiality and integrity, as well as weak access control. Since there is no destination address in a CAN frame, each node can send and receive messages that are broadcast to the bus based on predefined configuration. An adversary can easily sniff data flow and insert malicious messages, which poses a dangerous situation for both driver and passengers. Consequently, the protection of the CAN bus relies on proper authentication and encryption mechanism.

Several cryptographic protocol suites have been proposed for secure CAN bus communication [12–14]. The recently published cryptographic protocol suite is proposed by Palaniswamy et al. in 2020 [15]. This comprehensive protocol suite consists of seven protocols, covering the integrated process from session key establishment and update to data transmission and connectivity with external devices. We provide an informal analysis of this protocol suite and investigate its potential security weakness against suspension attacks. The analysis also identifies its redundancy during authentication and session key agreement. To address these drawbacks, we propose a scalable cryptographic protocol suite for CAN bus, providing several verified security properties and lower communication overhead against the large-scale intravehicular network.

The following are the main contributions of this paper:

(1) Through the analysis of the existing protocol suite, we identity the weakness against a new proposed attack, suspension attack. We raise a new protocol suite to mitigate the identified weakness based on the global counter scheme. Security verification is also presented in the AVISPA tool to this protocol suite.

(2) In order to construct a more scalable CAN protocol suite, we propose a broadcasting scheme in the initial session key distribution protocol (ISDP) by using the Chinese remainder theorem (CRT). Compared with the previous works, the new protocol reduces communication overhead significantly that removes the obstacles for supporting more ECU connections in one CAN bus.

(3) We propose an external device access protocol based on the certificateless signature scheme. Compared with certificate-based protocols, the proposed method lightens computational overhead and provides preferable efficiency.

The rest of the paper is organized as follows. Section 2 introduces the CAN bus protocol and intravehicular network, including the attack model and security requirements of this protocol suite. Section 3 presents various security solutions for CAN bus in prior works and the analysis of the existing protocol. Section 4 describes our new protocol suite. Section 5 verifies the security properties by using the AVISPA tool. Section 6 compares the simulation performance of our protocol suite with the existing schemes.

2. Background

2.1. CAN Network. The CAN protocol [16], designed by Bosch in 1981 for safety-critical and real-time applications, is a multimaster serial bus that has been widely used in industrial automation. CAN bus communication channel consists of twisted pair wires for differential signaling. The dominant level is represented by a logical 0, and the recessive level by a logical 1. Data frame, remote frame, error frame, and overload frame are four major frame types in CAN bus. Data frame, the most common message type, is used to transmit messages. ECU can also proactively request a message from others with matching identifiers using a remote frame. CAN protocol entitles all ECUs to send an error frame when an error is detected. Overload frame is used to provide for an extra delay between data or remote frames.

A standard CAN data frame can be separated into several parts as shown in Figure 1.

(i) *Start of Frame (SOF).* It is a single dominant bit informing the start of transmission when the bus is idle.

(ii) *Arbitration Field.* When multiple ECUs attempt to occupy the bus at the same time, an arbitration process is necessary to determine the sending sequence. ECU sending a higher identifier message will detect the lower identifier message and wait until the bus is free. Besides, identifier also implies the content of the message. For example, messages transmitting the temperature of a car engine have their own specific identifier.

(iii) *Control Field.* It contains one identifier extend (IDE) bit, one reserved bit always set to 0, and four bits as data length code (DLC).

(iv) *Data Field.* It refers to actual data of length from 0 to a maximum of 8 bytes for transferring information to another node.

(v) *CRC Field.* It consists of 15-bit cyclic redundancy code (CRC) and 1-bit CRC delimiter. If the CRC checksum of transmitted data is different from the calculated value of the recipient, a CRC error will be triggered.

(vi) *Acknowledge Field.* It consists of two bits such as ACK slot and ACK delimiter. By default, the sender node sets both of these two bits as recessive. If a receiver node verifies the message successfully, it replaces the ACK slot with dominant 0. Otherwise, a failed transmission will trigger an ACK response error and force the sending node to retransmit the message.

(vii) *End of Frame.* CAN frame is terminated by a flag consisting of seven recessive bits.

Typically, an intravehicular network is divided into three subnetworks: powertrain, body, and infotainment as shown in Figure 2. The powertrain subnetwork contains life-critical operations such as engine, brakes, and chassis control components, where high bandwidth and stable communication capabilities are available [1]. In the past decade, the infotainment subsystem containing entertainment and information functions is growing rapidly. The body subsystem generically controls the doors, seats, and rear with low-speed CAN. The communication of ECUs among these subnetworks is facilitated through gateway ECU (as well as other kinds of networks), which is assumed to be more powerful than the usual ECUs.

2.2. Attack Scenarios. Though attackers have numerous interfaces to invade the intravehicular CAN bus, we can conclude the existing attack scenarios specifically proposed

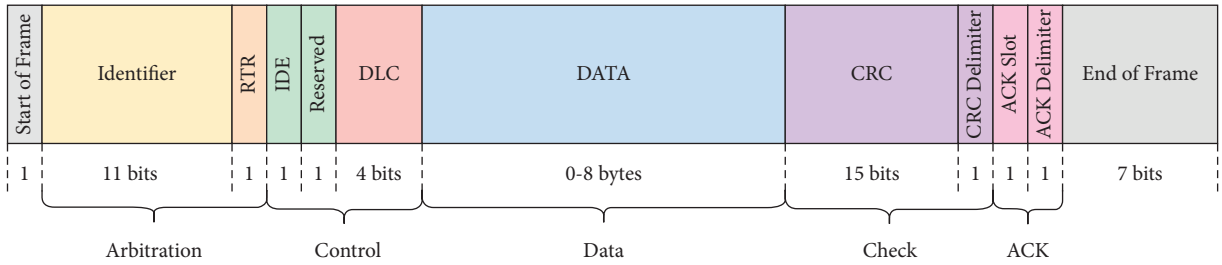

Figure 1: The structure of a standard CAN data frame.

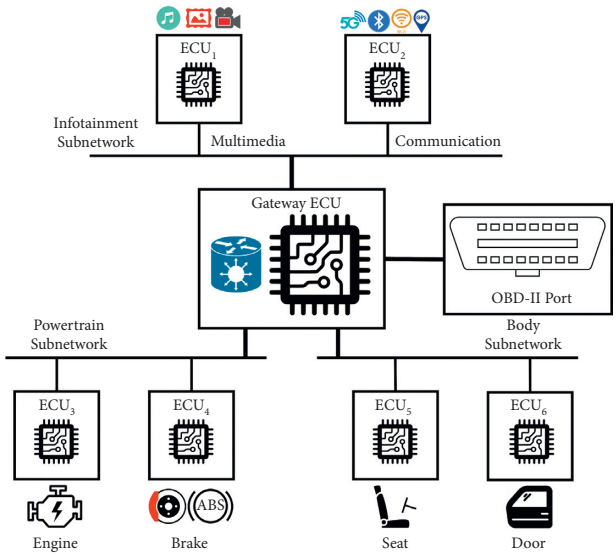

Figure 2: The topology of an intravehicle network.

by researchers into five parts: eavesdrop, replay, fuzzing, masquerade, and tampering [1, 17].

Eavesdrop. In this scenario, attackers have no ability to inject any messages into the bus. They can only monitor the communication on the bus and analyze the messages and flow as passive attackers.

Replay Attack. In a replay attack, there is no need for the attacker to understand the content of any message. The attacker just intercepts the historical message flow and records the corresponding relation between the message and the action caused by it in advance. Afterwards, the attacker can retransmit these messages at any time and obtain the same ECU's action as he expected. In fact, if the freshness of messages transmitted in the network cannot be guaranteed, the receiver of the replayed message will always be deceived.

Fuzzing Attack. The objective of a fuzzing attack is to override any periodic messages by sending fabricated messages with the same arbitration ID at a higher frequency. Thus, other nodes that normally receive messages are forced to receive the fuzzing attack messages more frequently than the legitimate ones.

Masquerade Attack. The objective of a masquerade attack is to inject illegal messages while concealing the fact that an ECU is compromised. To achieve this goal, the attacker needs to manipulate two ECUs, one original sender ECU and the other substitutional sender ECU. The attacker suppresses the original ECU from sending periodic messages and injects malicious messages at the original frequency with the substitutional sender ECU.

Tampering Attack. The objective of a tampering attack is to distort the messages in real-time, while the frame is transmitting. The attacker first launches a bus-off attack [17] to force the victim ECU to enter "error passive" state in advance. Afterwards, the attacker can change the transmitting data field arbitrarily without triggering any bit errors. The CRC field is also changed accordingly to prevent CRC error.

Suspension Attack. We also identified a new attack scenario, suspension attack, which is effective on existing CAN cryptographic protocol suites. This attack can be launched by leveraging the error handling mechanism and the lack of data flow sequence check. For example, in [13, 15], counters are assigned to synchronize messages between sender ECU and receiver ECU for each ID. However, the cross-ID sequence cannot be protected by using the separated ID-based counter.

Here, we propose two ways for attackers to launch a suspension attack.

First, attackers can send an error frame when the sending node is transmitting the ACK or SOF field of the proposed data frame. Since the data and CRC fields are completed, attacks can sniff the whole message while concealing its availability with an error frame. Once other ECUs receive the error frame and drop the received data frame, their counter will not update as normal, which gives a chance for attackers to replay this message at any time afterwards.

Second, attackers can just suspend one ECU when it is ready to send a new data frame. Before the message is sent, this ECU will be shut down, and this message will be cached in the transmission buffer. Since the message has never been sent out, the counter will not be overdue unless other ECUs send a new message with the same ID.

We used Raspberry Pi and CAN analyzer to build a demonstration system to present a suspension attack as shown in Figure 3. In our system, the attacker is assumed to have the ability to block the messages in the sending buffer of a victim sender ECU. Thus, the attacker can launch a suspension attack as long as he refuses to forward the message

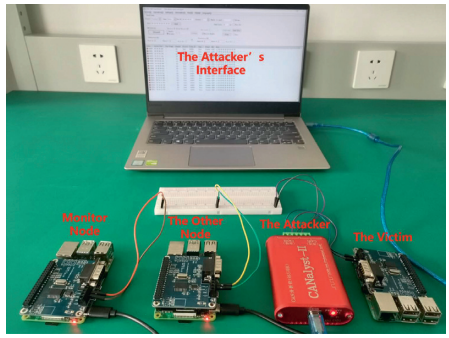

Figure 3: The demonstration system for suspension attack.

to the CAN bus and replays it afterwards. The result shows that the delayed message cannot be detected by other ECU nodes, which implies vulnerability in protocols using a local counter scheme.

2.3. Security Requirements. Based on the attack scenarios presented above, we give an attack model to describe the capability of the adversary in a formal way. The attacker is assumed to run the efficient polynomial-time algorithm, which has full control to the intravehicle network. The attacker is under the Dolev–Yao model [18], with the capabilities to act as a legitimate user to send, obtain, or even manipulate messages from any party in the interactive cryptographic protocol. The attacker can intercept and masquerade a legitimate user to interact with other parts in the protocol, attempting to inject malicious messages or steal ephemeral and long-term secrets. In Section 5, a security verification based on AVISPA gives an attacker instance under Dolev–Yao model to this proposed cryptographic protocol suite. In Section 4.7, we strengthen the capabilities of the attackers just like under Canetti–Krawczyk model and analyze the protocol informally.

Under the presence of the adversary defined above, the following security goals are going to be of interest:

(i) *Confidentiality.* In the original CAN protocol, data frames are transmitted in clear, making the key information such as ID and data be exposed to an attacker apparently. Accordingly, every data frame should be encrypted to avoid eavesdropping. Only a legitimate node with a proper key can decrypt the message.

(ii) *Mutual Authentication.* Two communicating entities must confirm the legality of the other before transmitting, in case of temporarily illegal access to steal traffic.

(iii) *Key Freshness.* The session key should be updated in time under the circumstances when a counter overflow happens or an external device connection is released, which guarantees that the current session key is randomly and independently selected.

(iv) *Forward and Backward Secrecy.* Forward secrecy protects past sessions against future compromises of keys. That is, the disclosure of the current session key will not lead to the leak of the session key that has been used in the past [19]. Backward secrecy indicates that compromise of past session keys does not affect the security of future session keys.

3. Related Work

Since confidentiality and authenticity of the messages transmitted on the CAN bus are not guaranteed, attackers have numerous opportunities to remotely eavesdrop or tamper with data and cause great danger to the safety of vehicles. In order to solve the potential safety problems of the intravehicular CAN network, researchers have proposed various types of solutions.

In order to reduce computational complexity and bandwidth on the CAN bus, several lightweight cryptographic protocols have been proposed for confidentiality, integrity, and authenticity. Nilsson et al. proposed a delayed data authentication mechanism, which uses 64 bit MAC for four data frames embedded in the following four data frames with no data [20]. Herrewege et al. proposed a backward-compatible data authentication protocol based on HMAC [21]. Groza et al. designed LiBrA-CAN, an efficient protocol based on mixed message authentication codes (M-MACs), which aggregates several MACs into one to increase security [12]. Subsequently, Groza et al. proposed a TESLA-like [22] protocol by the priority of messages [23]. They used a master node to verify messages from the sender ECU and recomputed the tag with the key shared with the receiver ECU. Farag et al. proposed CANTrack, an intuitive scheme based on a dynamic key for encrypting CAN messages to prevent replay attacks [24]. Fassak et al. proposed a secure protocol for ECU authentication and session key establishment based on elliptic curve cryptography (ECC) [25]. Pan et al. proposed a new security scheme combining private key derivation algorithm based on electronic fingerprint and ECC algorithm for shared key distribution, which enhanced the nonreproducibility of messages [26].

Other methods improve the security of CAN bus in physical characteristics or other aspects. Lin et al. presented a new formulation that models the path-based security constraints and minimizes security risk directly to solve the problem that the overhead of security mechanisms might cause violations of design constraints [27]. Jain et al. utilized the physical properties of the CAN bus to construct a tree-based group key exchange protocol, which has logarithmic complexity for node addition and deletion [28]. Nürnberger and Rossow proposed VatiCAN, which was designed to be backward-compatible to allow tried and trusted components to rely on the same CAN messages [29]. Humayed and Luo proposed an ID-hopping scheme aiming to prevent targeted DoS attacks [30]. Siddiqui et al. proposed an authentication framework using the physical unclonable function for enhanced security, which can be integrated with existing resource constraint embedded devices [31].

The recent CAN security scheme based on the cryptographic protocol is proposed by Palaniswamy [15]. They analyzed the existing frame-level authentication protocol proposed by Woo et al. [13] and identified weaknesses and

limitations. The enhanced protocol suite covers entity authentication, secure transmission for remote frames and data frames, session key updates, and secure access with external devices. After evaluation, the new protocol suite is claimed to satisfy known key secrecy (KKS) [32] and withstand the attacks, namely masquerading, replay, and MITM. However, the protocol suite proposed by Palaniswamy et al. still has some drawbacks in efficiency and potential security weaknesses. Initial session key distribution protocol (ISDP) and session key update protocol (SKUP) are derived from a variant of the AKEP2 protocol. Despite security is guaranteed, broadcasting property in CAN bus is not leveraged to these protocols. GECU has to establish redundant challenge-response authentication with each ECU and cause unnecessary communication overheads. As for data transmission protocol (DTP) and RTRP protocol, each counter is arranged to record the data flow of a certain arbitration ID. Unfortunately, this scheme is vulnerable to suspension attacks introduced in Section 2.2. When a suspension attack is launched, messages will be forced to delay without disturbing their consolidated sequence, so this attack will not be defended by detecting the anomalous counter value in HMAC. The other drawback is that no protocols have been designed for cross-subsystem data transmission. Since there are usually two or more CAN bus in modern vehicles, it is necessary to directly exchange cross-subsystem messages encrypted with different session keys. In vehicle connection protocol (VCP), the digital certificate introduced in [13, 15] will lead to extra computational overheads in complicated certificate management and verification procedure of certificate chains. These drawbacks motivate us to improve its performance and reconstruct the whole protocol suite.

4. Scalable Protocol Suite

In the new proposed protocol suite, we reconstruct the design of previous works by enhancing the efficiency during authentication procedure and addressing security weaknesses against local counters. The new protocol suite consists of six phases, among which ISDP and SKUP are constructed for session key derivation; DTP, RTRP, and cross-subsystem data transmission protocol (CSTP) are designed to protect confidentiality and integrity for data transmission; and VCP is for external device authentication and key establishment. The notations used in the proposed protocol suite are presented in Table 1.

4.1. Initial Session Key Distribution Protocol. This protocol allows GECU to establish an initial session key with ECUs in the same subnetwork as shown in Figure 4. The existing ISDP protocols are based on some variants of the AKEP2 protocol, requiring GECU to finish the complete authentication process with each ECUs [13, 15]. We propose a simplified ISDP protocol by fully leveraging the broadcast mechanism of the CAN bus and the property of the Chinese remainder theorem. In this protocol, GECU completes authentication and session key distribution with all the ECUs in the same time.

Step 1. GECU Challenge

The protocol is launched by GECU. GECU generates a random integer Seed and uses long-term preshared key GK to calculate its HMAC value $\text{MAC} = H_{GK}(ID_{\text{GECU}} \| \text{Seed})$, then pack the seed and HMAC together, and send to the bus.

Step 2. ECU Response

Each of ECU_i can verify the Seed and generate another random integer R_i and random prime Pi as challenge values. Then ECU_i generates $\text{MAC}_i = H_G K(K_i \| \text{Seed} \| R_i \| P_i)$ as a response for GECU's challenge. K_i, R_i, and P_i are also contained in MAC_i for identification and message integrity.

Step 3. GECU Response

After receiving ECUs' response, GECU separately verifies MAC_1 to MAC_n and generates $\text{Hash}_i = H_{GK}(K_i \| \text{Seed} \| R_i)$ as response value for ECU_i which is slightly different from MAC_i. By using the property of the Chinese remainder theorem, GECU can construct $S = (\sum_{i=1}^{n} \text{Hash}_i y_i Z_i) \text{mod} Z$ that is congruent to Hash_i modulo P_i for each i from 1 to n. Therefore, each ECU_i can verify their unique response Hash_i using the same S. Finally, session keys EK and AK can be derived from i for both GECU and ECUs.

4.2. Data Frame Transmission Protocol. The security goals of data frame transmission protocol are providing confidentiality and integrity for CAN data frame. To prevent replay attack against data frame, a counter is needed to provide the freshness of ciphertext as shown in Figure 5. ECUs in the same subnetwork share a global counter.

Step 1. Data Frame Generation

When an ECU receives a message from the bus, it always increases its counter to guarantee synchronization. In DTP protocol, the sender ECU uses the CTR mode of AES to encrypt the message. Since the length of the data field is 8 bytes, only the first 64 bits of $\text{Enc}_{EK}(CTR)$ are used to generate ciphertext $\text{Enc}_{EK}(CTR) \oplus m$. The authentication part consists of HMAC of arbitration ID, ciphertext, and counter.

Step 2. Decryption/Counter Update

After the receiver ECU receives a message, HMAC is verified first before decryption. Other ECUs also increase their counters to guarantee the synchronization except an error frame is received.

4.3. Remote Frame Transmission Protocol. RTRP protocol is similar to DTP as shown in Figure 6. The main difference is that the remote frame does not have a data field; thus, only hash computation is necessary.

Step 1. RTR Frame Generation

When the receiver ECU needs messages from the sender ECU, the former ECU sends a remote frame

Table 1: Notations used in the proposed protocols.

Notation	Description
GECU	Gateway ECU
ECU_i	ECU using identity i
ID_i	Arbitration identifier used by the i^{th} ECU
$Seed_k$	Seed value to derivate session key in k^{th} session
R_i	Random value generated by ECU_i
GK	Long-term preshared symmetric key between GECU and all ECUs
K_i	Long-term preshared symmetric key between GECU and ECU_i
P_i	Big prime generated by ECU_i
S	Aggregated HMAC value generated by *GECU*
KDF_k	Session key derivation function
EK_k	Encryption key of k^{th} session
AK_k	Authentication key of k^{th} session
CTR_i	Frame counter of i^{th} subnetwork
Q	Generator on the elliptic curve group $\mathbb{G}$ with order q
a, b, r_i	Random number generated in Z_q
(y_i, Y_i)	Secret key and public key pair used in external connection for device i
(s, P_{pub})	Secret key and public key pair of key generation center
$\text{Enc}_{\text{EK}}(\cdot)$	Symmetric encryption function
$H_k(\cdot)$	Secure keyed hash function, (y_i, Y_i): $\{0,1\}^* \times \text{key} \longrightarrow \{0,1\}^{32}$
$H_1(\cdot,\cdot)$	Secure hash function, $H_1(\cdot,\cdot)$: $\{0,1\}^{32} \times \mathbb{G} \longrightarrow \{0,1\}^{32}$

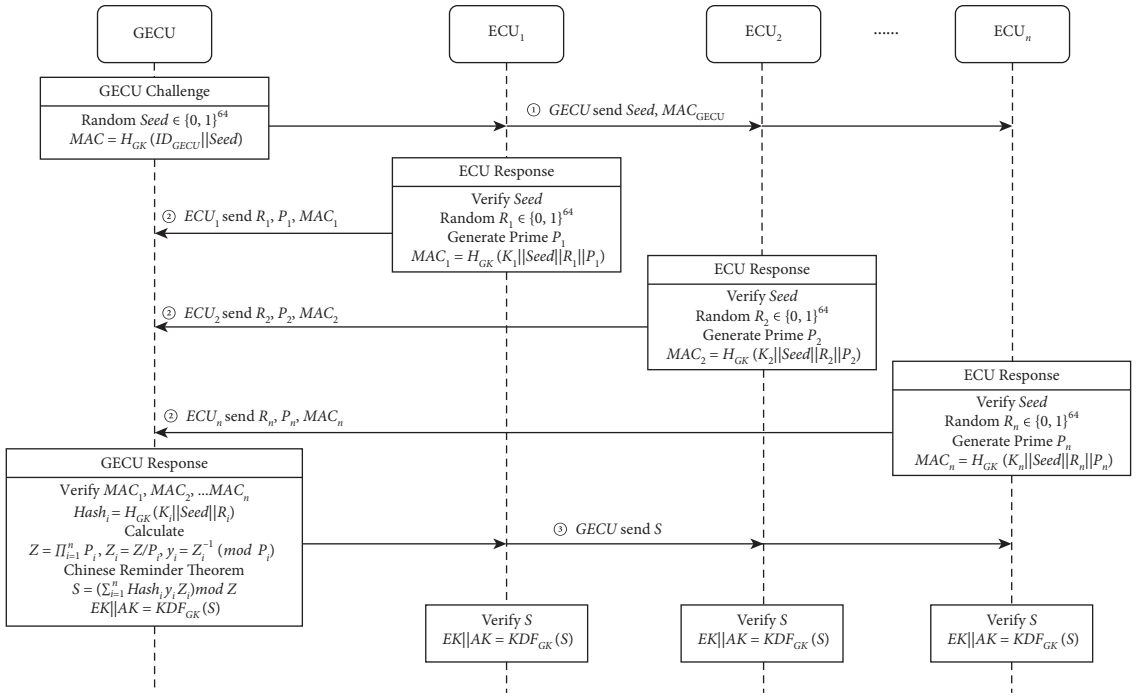

Figure 4: The procedure of ISDP.

$\text{MAC}_{\text{RTR}} = H_{AK}(\text{RTR Frame} \| ID_r \| \text{CTR})$ containing arbitration ID and counter value.

Step 2. Verification/Counter Update

After the sender ECU receives the message, it increases its counter if the MAC is verified. Similarly, other ECUs also update their counters if the message is received. Afterwards, the sender ECU can send data frames following the DTP protocol.

4.4. Cross-Subsystem Data Frame Transmission Protocol. CAN buses in in-vehicle networks are separated into different subnetworks, such as powertrain, chassis, safety, and infotainment parts. All subnetworks are independent of each other but connected with a center gateway. However, the demand for cross-subnetwork data transmission still exists. In ISDP protocol, session keys are established in their own subnetworks. ECUs in different subnetworks cannot communicate directly. Therefore, a cross-data transmission

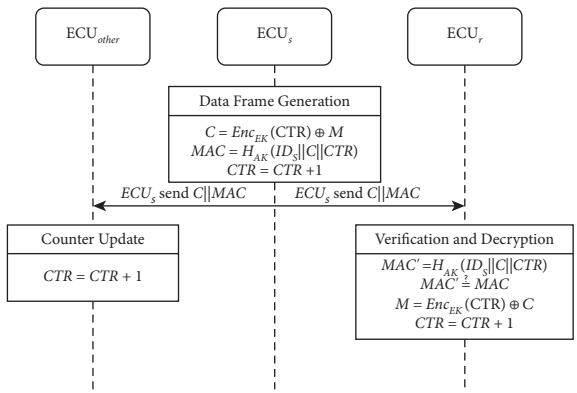

Figure 5: The procedure of DTP.

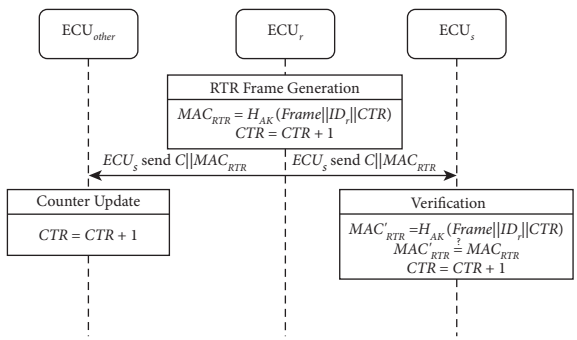

Figure 6: The procedure of RTRP.

protocol is needed for encryption communication. In addition, synchronization of counters in different subnetworks is impossible. Thus, message retransmission of GECU is essential for cross-subnetwork messages as shown in Figure 7.

Step 1. Data Frame Generation

At first, the sender ECU from subnetwork i generates a data frame using EK and AK as in DTP protocol.

Step 2. Verification and Re-encryption

When GECU receives a frame with a certain arbitration ID, a retransmission mechanism will be triggered. GECU verifies the messages, re-encrypts with the session key in subnetwork j and re-transmits them to the other subnetwork following a routing table. GECU can convert counter values from different subnetworks and provide synchronization.

Step 3. Verification and Decryption

Finally, the receiver ECU can receive and verify the messages from the other subnetwork.

4.5. Session Key Update Protocol. Session keys need to be updated under two conditions. First, after a predefined time, counters need to be reinitialized in case of overflow. Second, when external devices are released from the vehicle, the session key will be updated in case of

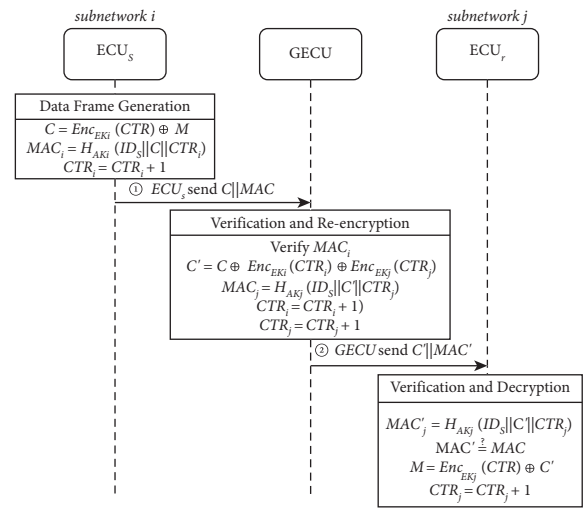

Figure 7: The procedure of CSTP.

malicious leakage. The procedure of SKUP is shown in Figure 8.

Step 1. New Session Key Derivation

Session key update protocol is launched by GECU. When the global counter reaches a predefined value or releases an external device, GECU will send a data frame. The data field is a new random seed, and the authentication part is the HMAC value $\mathrm{MAC} = H_{AK}(ID_{\mathrm{GECU}} \| \mathrm{Seed}_{k+1} \| GK)$ that contains arbitration ID, a fresh random seed, and long-term key GK to ensure authenticity.

Step 2. Verification and Update

All other ECUs will verify this message and derive a new session key from this HMAC value. Then, ECUs' responses will be sent to achieve the key confirmation.

4.6. Vehicle Connection Protocol. When an external device is connected to the CAN bus, the authentication and key agreement process must be done with GECU. However, certificate-based authentication is heavy for vehicular access protocol that needs the support of public key infrastructure (PKI). To avoid the usage of certification, we propose a cryptographic scheme that the pair of secret and public keys can only be derived by an authority (e.g., the car manufacturer), which can ensure the creditability of the vehicle's public key as shown in Figure 9.

Step 0. Initial Key Generation

When a vehicle or an external device is manufactured, it should be registered with key generation center (KGC) that holds the master secret key s and publishes its public key $P_{\mathrm{pub}} = s \cdot Q$ as the public parameter. KGC derives a pair of asymmetric keys (y_i, Y_i) for the registered device, where $Y_i = r_i \cdot Q$, $y_i = r + s \cdot H_1 (ID_i, Y_i)$, and r_i is random generated in Z_q. Subsequently, (y_i, Y_i) is injected to the memory of the device via a secure channel.

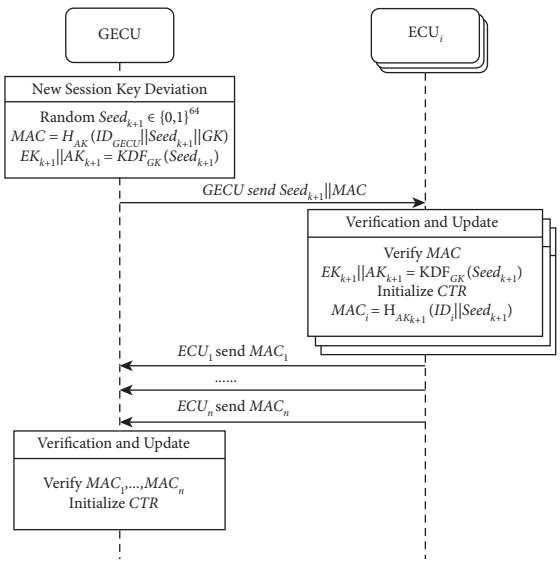

Figure 8: The procedure of SKUP.

Step 1. EDEV Challenge

When the external device is intended to connect to a vehicle, it first generates a random challenge value $V_E = a \cdot Q$.

Step 2. GECU Response and Challenge

Then GECU generates a signature $S_G = b + y_G \cdot h_{E2}$ on the identity ID_E and challenge value V_E of the external device. Then the signature, challenge value $V_G = b \cdot Q$, and public key Y_G are transmitted together.

Step 3. EDEV Response

The external device can verify the signature S_G with GECU's public key Y_G. Accordingly, the external device generates a signature $S_E = a + y_E \cdot h_{G2}$ on GECU's identity and challenge value. Afterwards, the signature S_E is transmitted with public key Y_E. Since GECU has been authenticated by the external device, the session key can be derived by the ECDH scheme on the external device side.

Step 4. GECU Verification

GECU can authenticate with the external device if $S_E P = V_E + h'_{G2}(Y_E + h_{E1} P_{\text{pub}})$ is satisfied. By using the ECDH scheme, the session key can be derived on the GECU side.

4.7. Informal Security Analysis of New Protocol Suite

Mutual Authentication. In the ISDP protocol, authentication relies on the preshared symmetric key. The responses generated by both ECU and GECU require a group key GK shared among entities in a subnetwork and a unique key K_i shared between GECU and ECU$_i$, which are unobtainable to the adversary. In VCP protocol, asymmetric key pair held by the external device and GECU guarantees authenticity.

Session Key Agreement. In ISDP protocol and SKUP protocol, the session key is derived from a group broadcasted verification value and a preshared key GK. In VCP protocol, the session key is derived from Diffie–Hellman scheme with signature for entity authentication, which avoids the threat of man-in-the-middle attack.

Protocol Attack Resistance

(1) *Replay Attack*. If attackers act as GECU to launch ISDP protocol by replaying the initial challenge message, the impersonated GECU cannot generate a valid response without GK. Similarly, attackers cannot generate a valid ECU response message as normal ECU because of the freshness of the seed. In DTP and RTRP protocol, by using a counter for synchronization, an old counter value from a replay message will not be accepted by normal receiver ECU. In SKUP protocol, the new session key derivation message is the last frame using the current session key, which is meaningless to replay. In VCP protocol, both GECU and the external device must generate a signature to a fresh unique value from the other side, which is impossible to replay.

(2) *Suspension Attack*. The new proposed DTP, CSTP, and RTRP protocol uses a global counter for all messages with a different ID. Suspension to any random ECU will lead to desynchronization of the delayed message and malicious messages will be discarded automatically.

(3) *Masquerade Attack*. When the attacker tries to send a malicious random data frame, the receiver ECU or GECU will detect the anomalies and send an error with the protection of frame authentication provided by HMAC. Other ECUs on the bus will not update their counter. Therefore, an injection attack will not influence the availability of the protocol through counter synchronization mechanism.

Forward/Backward Secrecy. SKUP provides forward/backward secrecy. Since GK is contained in the calculation of MAC, even if AK is revealed to the attacker, EK and AK used in other sessions will not be exposed. In other words, only for entities with the knowledge of both AK and GK, SKUP can be triggered validly.

5. Security Verification

AVISPA is a push-button tool for formal analysis of the security protocol. It provides a modular and expressive formal language (HLPSL) for specifying protocols and their security properties, integrating different back ends that implement a variety of automatic analysis techniques. To generate the message sequence chart and check that the HLPSL specification is correct, we used the SPAN tool to choose the automatic analysis techniques and run the HLPSL script. AVISPA comprises four back ends: OFMC, CL-AtSe, SATMC, and TA4SP. Because of the calculation

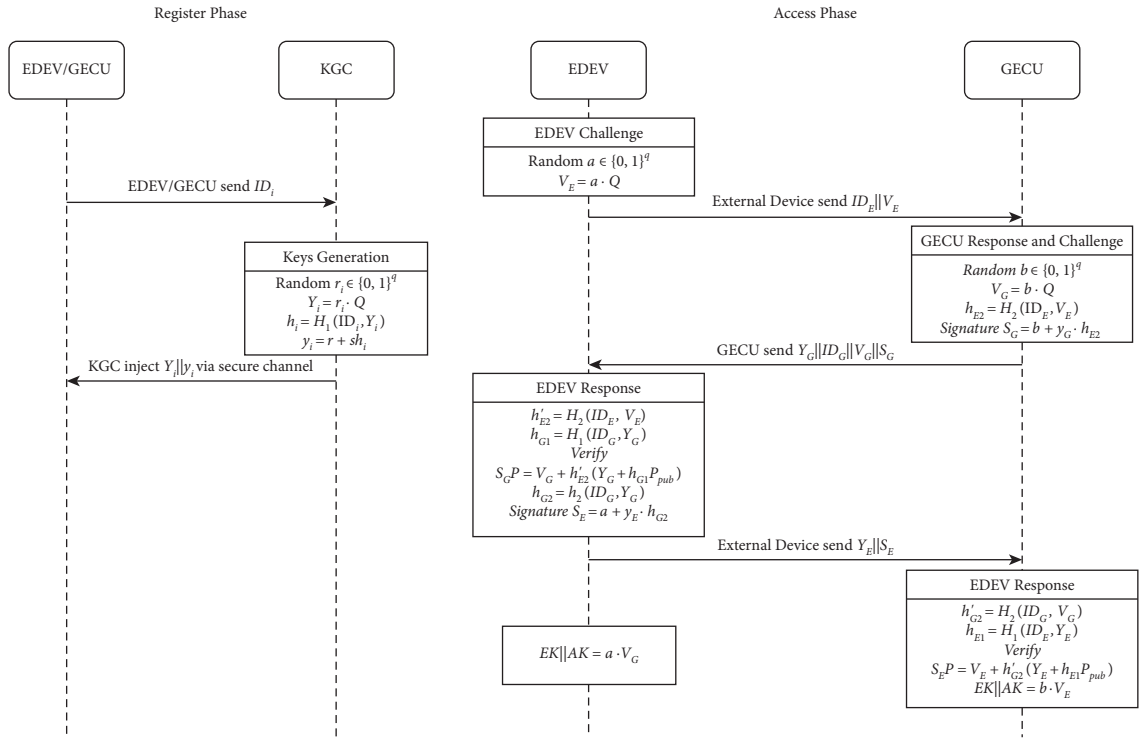

Figure 9: The procedure of VCP.

process of the security protocol, we used the OFMC tool to test the security properties.

We begin with the description of ISDP in HLPSL. Figure 10 illustrates the information and actions of ECU_i in ISDP protocol. This element just illustrates the basic roles of the protocol, namely ECU and GECU. Preknown information is set by the parameters of the role. Variables described in section "local" are used to create or receive information. Section "transition" describes a detailed protocol process of ECU_i. "State" is used to identify the status of the role and facilitate the next action. Security properties are also described in section "transition" that will be analyzed with the implementation of the protocol.

The second element is shown in Figure 11. This element illustrates the composition of those basic roles for representing the protocol to facilitate the reference of the environment. Instead of section "transition," section "composition" refers to basic roles and transfers the preknown information to the basic roles.

Figure 12 illustrates the execution environment of the protocol instance to study. To instantiate the protocol, this element describes the preknown information of basic role and role sessions. The set "intruder_knowledge" indicates the information that the intruder can obtain. Section "composition" refers to the role session for the instance of the protocol. If multiple sessions are allowed to occur at the same time, in reality, this section can refer to several sessions to excavate possible attacks.

Figure 13 includes two elements: section "goal" and the execution of role environment. Section "goal" illustrates the declaration of the security properties to analyze. The execution of environment starts the process of the protocol and analyzes the proposed security properties.

Descriptions for other protocols with their own security properties defined in AVISPA are similar to ISDP protocol. Table 2 shows the total verification results of the proposed protocol suite in AVISPA. The confidentiality and authenticity of critical parameters in each of the six protocols are all verified successfully. Besides, forward secrecy is also guaranteed in ISDP, SKUP, and VCP protocol.

6. Performance Analysis

This section presents a performance comparison of our protocol suite with the protocols of Woo et al. [13] and Palaniswamy et al. [15]. In the comparison, we consider security properties as well as costs in communication and computation. A simulation in MATLAB is also presented for efficiency comparison. The result shows that the proposed scheme achieves the minimum communication complexity in ISDP/SKUP protocol and the best computation cost in the VCP protocol. Furthermore, enhanced security attributes are also provided in this scheme that is not available in the previous works.

Table 3 shows the communication efficiency in the session key establishment stage, covering ISDP and SKUP protocols. As for ISDP protocol, the former designs in [13, 15] request GECU to establish a session key with each ECU independently. In other words, GECU needs to accomplish a mutual challenge-response scheme with each ECUs in sequence, which leads the message complexity up to $3n$, where n represents the number of ECU nodes. In our modified ISDP design, by using the CRT theorem, GECU's

```
role role_ECUi(
  Ei,Ej,G       : agent,
  Ki,GK         : symmetric_key,
  IDi,IDj,IDgecu : text,
  KDF,H         : hash_func,
  SND,RCV       : channel(dy))
played_by Ei def=
  local
      State       :nat,
      Rj,Ri,Seed,Pi   :text,
      EK,AK       :hash(symmetric_key.message),
      MACg        :hash(symmetric_key.text.text),
      MACi        :hash(symmetric_key.symmetric_key.text.text.text),
      MACj        :hash(symmetric_key.symmetric_key.text.text.text),
      MAC         :hash(symmetric_key.hash(symmetric_key.symmetric_key.text.text)
                    .hash(symmetric_key.symmetric_key.text.text).message),
      Hashi       :hash(symmetric_key.symmetric_key.text.text),
      Hashj       :hash(symmetric_key.symmetric_key.text.text),
      S           :message
  init
    State := 0
  transition
  1. State = 0 /\ RCV(Seed'.MACg') /\ MACg' = H(GK.IDgecu.Seed') =|>
     State' := 2 /\ Ri' := new() /\ Pi' := new() /\ MACi' := H(GK.Ki.Seed'.Ri'.Pi') /\
                    SND(Ri'.Pi'.MACi') /\ witness(Ei,G,e_g_MACi,MACi')

  2. State = 2 /\ RCV(Hashi'.Hashj'.S'.MAC') /\ MAC' = H(GK.Hashi'.Hashj'.S') /\
                  Hashi' = H(GK.Ki.Seed.Ri)  =|>
     State' := 4 /\ EK' := KDF(GK.S') /\ AK' := KDF(GK.S') /\
                    SND(Hashi'.Hashj'.S'.MAC') /\
                    request(Ei,G,e_g_MACgi,MACg) /\
                    request(Ei,G,e_g_S,S') /\ witness(Ei,Ej,e_e_S,S') /\
                    secret(EK',ek1,{Ei,Ej,G}) /\ secret(AK',ak1,{Ei,Ej,G})

end role
```

Figure 10: Role role_ECUi of ISDP in HLPSL.

```
role session(
  Ei,Ej,G       :agent,
  Ki,Kj,GK      : symmetric_key,
  IDi,IDj,IDgecu : text,
  KDF,H,Mutiple,Division,Add,Invert,Mod        : hash_func)
def=
  local SND,RCV  : channel(dy)

  composition
    role_ECUi(Ei,Ej,G,Ki,GK,IDi,IDj,IDgecu,KDF,H,SND,RCV)
          /\ role_ECUj(Ei,Ej,G,Kj,GK,IDi,IDj,IDgecu,KDF,H,SND,RCV)
          /\ role_GECU(Ei,Ej,G,Ki,Kj,GK,IDi,IDj,IDgecu,KDF,H,Mutiple,
             Division,Add,Invert,Mod,SND,RCV)

end role
```

Figure 11: Role session of ISDP in HLPSL.

```
role environment()
def=
  const
    ek1,ek2,ek3,ak1,ak2,ak3                    : protocol_id,
    e_g_MACi, e_g_MACj                         : protocol_id,
    e_g_MACgi, e_g_MACgj                       : protocol_id,
    e_g_S, e_e_S                               : protocol_id,
    ki,kj,gk                                   : symmetric_key,
    ecui,ecuj,gecu                             : agent,
    idi,idj,idgecu,s2                          : text,
    kdf,h,mutiple,division,add,invert,mod      : hash_func

  intruder_knowledge = {idi, idj, idgecu, ecui, ecuj, gecu, kdf, h,
                        mutiple, division, add, invert, mod, kdf(s2.gk)}

  composition
    session(ecui,ecuj,gecu,ki,kj,gk,idi,idj,idgecu,kdf,h,mutiple,division,
            add,invert,mod)

end role
```

Figure 12: Role environment of ISDP in HLPSL.

```
goal
  secrecy_of ek1
  secrecy_of ek2
  secrecy_of ek3
  secrecy_of ak1
  secrecy_of ak2
  secrecy_of ak3
  authentication_on e_g_MACi
  authentication_on e_e_S
  authentication_on e_g_S
  authentication_on e_g_MACj
  authentication_on e_g_MACgi
  authentication_on e_g_MACgj
end goal

environment()
```

Figure 13: Goal and execution of environment of ISDP in HLPSL.

response to each ECUs can be aggregated in a single message and decrease the communication complexity significantly to $n + 1$. Furthermore, in modified ISDP protocol, every ECUs receive GECU's challenge at the same time; therefore, their calculation can be processed simultaneously, which is beneficial to lighten computation delay. As for the SKUP protocol, compared with the design of Woo et al. [13], we simplified the in-sequence update scheme by using a broadcast scheme like the ISDP protocol.

In this section, we compare the computation overhead in VCP protocol with other current schemes. Since certificate verification operation is composed of two point multiplication operations, one point addition operation, and one hash operation, we mainly involve the time of hash operation, point multiplication over the elliptic curve (EC), and so on. The time corresponding to each operation is listed in Table 4 according to [33]. Table 5 shows the computation comparison of different schemes. The existing schemes like that in [13, 15] use PKI and digital certificates to provide a trusted public key that leads to extra computation for certificate verification. Due to the certificateless authentication signature scheme, our new proposed protocol achieves the minimum computation overhead.

The new proposed protocol suite shows not only efficiency in communication and computation aspects but also rich security features. Table 6 shows evidence of enhanced security properties especially in resistance to suspension attack, which is discussed in Section 2.2. Based on Tables 3, 5, and 6, the proposed scheme is efficient and robust in security than the previous works.

We simulate message communication delay for session key establishment procedure using Windows 10

Table 2: Result of verification.

Protocol	Property	Result	Attack
ISDP	EK Confidentiality	Pass	None
	AK Confidentiality	Pass	
	MAC Authentication	Pass	
	MAC_i Authentication	Pass	
	MAC_j Authentication	Pass	
	Forward secrecy	Pass	
DTP	CTR Confidentiality	Pass	None
	M Confidentiality	Pass	
	MAC Authentication	Pass	
RTRP	CTR Confidentiality	Pass	None
	MAC Authentication	Pass	
CSTP	CTR Confidentiality	Pass	None
	M Confidentiality	Pass	
	MAC Authentication	Pass	
	MAC′ Authentication	Pass	
SKUP	EK Confidentiality	Pass	None
	AK Confidentiality	Pass	
	MAC_i Authentication	Pass	
	MAC_j Authentication	Pass	
	Forward secrecy	Pass	
VCP	a Confidentiality	Pass	None
	b Confidentiality	Pass	
	EK Confidentiality	Pass	
	AK Confidentiality	Pass	
	V_E Authentication	Pass	
	V_G Authentication	Pass	
	Forward secrecy	Pass	

Table 3: Communication costs in the key establishment procedure.

	Woo et al. [13]	Palaniswamy et al. [15]	Our scheme
Communications involved in the ISDP protocol			
ECU	1	1	1
GECU	2	2	2
Message complexity	$3n$	$3n$	$n+1$
Communications involved in the SKUP protocol			
ECU	1	1	1
GECU	2	2	1
Message complexity	$2n$	$n+1$	n
Total message complexity	$5n$	$4n+1$	$2n+1$

Table 4: Computation costs in VCP protocol.

Operation	Notation	Time (μs)
Hash function	T_{hash}	67
Symmetric encryption	T_E	161
Multiplication over EC	T_{ECM}	612
Addition over EC	T_{ECA}	125

Table 5: Computation costs in VCP protocol.

Schemes	Overall computation cost
[13]	$9T_{\text{ECM}} + 4T_{\text{ECA}} + 6T_H + T_E \approx 6571\,\mu$s
[15]	$9T_{\text{ECM}} + 4T_{\text{ECA}} + 8T_H + T_E \approx 6705\,\mu$s
Our scheme	$8T_{\text{ECM}} + 4T_{\text{ECA}} + 8T_H \approx 5932\,\mu$s

environment with Intel Core i5-8265U @1.6 GHz and 8 GB RAM in MATLAB 2019b, involving the execution of ISDP and SKUP protocols. We set four scenarios for evaluating the communication delay with the number of ECUs in CAN bus with different CAN bus speeds. In the presented four scenarios, we, respectively, set 50, 75, 100, and 125 ECU devices in the CAN bus. As shown in Figure 14, message delay increases in all of the three schemes, but the message delay in our method is distinctively lower than the other two. Also, our scheme shows more competitive performance in large ECU numbers due to the low message complexity, which implies a broad prospect, especially in large-scale intravehicular networks.

Table 6: Security attributes comparison.

	[13]	[15]	Our scheme
Mutual entity authentication	✓	✓	✓
Session key freshness	✓	✓	✓
Resistance to DoS attack	×	✓	✓
Formal verification	×	✓	✓
Forward secrecy	×	✓	✓
Certificate free	×	×	✓
Resistance to suspension attack	×	×	✓

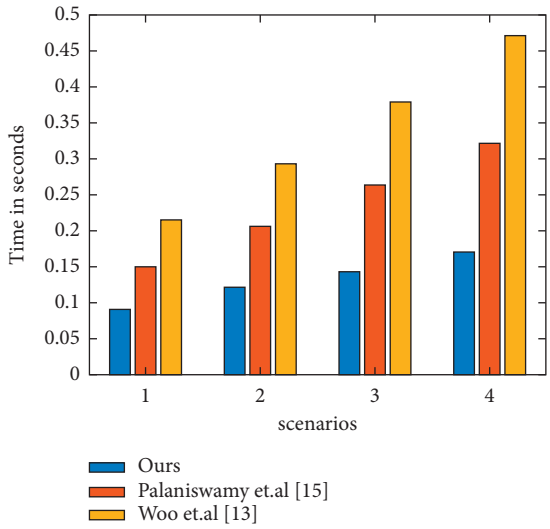

Figure 14: MATLAB simulation for communication delay.

7. Conclusion

This paper analyzed the limitation of the state-of-the-art cryptographic protocol suite for intravehicular CAN network by presenting its security weakness against our new identified CAN attack scenario, suspension attack. A new protocol suite is proposed to overcome the vulnerability as well as increase efficiency. The broadcasting scheme in the key distribution phase and certificateless schemes used in the external access phase have shown the ability in reducing communicational and computational overhead in performance analysis. Several security properties of the proposed protocol suite are also convinced under the security verification in AVISPA. In the future, we will enhance the secure CAN protocol suite by associating with intrusion detection systems to resist a broader range of attacks such as denial of service and extending the secure protocol to other intravehicular bus networks such as FlexRay and MOST to give an integrated solution for vehicle bus security system.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that no conflicts of interest exist.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (32071775) and the Opening Project of Shanghai Trusted Industrial Control Platform.

References

[1] T. Huang, J. Zhou, Y. Wang, and A. Cheng, "On the security of in-vehicle hybrid network: status and challenges," in *Proceedings of the International Conference on Information Security Practice and Experience*, pp. 621–637, Springer, Melbourne, Australia, December 2017.

[2] A. B. E. S. E. Team, "Future Advances in Body Electronics," Technical Report, NXP Semiconductors, Eindhoven, Netherlands, 2017.

[3] K. Koscher, S. Savage, F. Roesner et al., "Experimental Security Analysis of a Modern Automobile," in *Proceedings of the 2010 IEEE Symposium On Security And Privacy*, pp. 447–462, IEEE Computer Society, Oakland, California, May 2010.

[4] I. Rouf, R. D. Miller, H. A. Mustafa et al., "Security and privacy vulnerabilities of in-car wireless networks: a tire pressure monitoring system case study," *USENIX Security Symposium*, vol. 10, 2010.

[5] C. Miller and C. Valasek, "A survey of remote automotive attack surfaces," *Black Hat USA*, vol. 201494 pages, 2014.

[6] S. Checkoway, D. McCoy, B. Kantor et al., "Comprehensive experimental analyses of automotive attack surfaces," in *Proceedings of the USENIX Security Symposium*, vol. 4, pp. 447–462, San Francisco, CA, USA, August 2011.

[7] I. Foster, A. Prudhomme, K. Koscher, and S. Savage, "Fast and vulnerable: A story of telematic failures," in *Proceedings of the 9th USENIX Workshop on Offensive Technologies WOOT 15)*, Washington, DC, USA, August 2015.

[8] C. Miller and C. Valasek, "Remote exploitation of an unaltered passenger vehicle," *Black Hat USA*, vol. 2015, 2015.

[9] C. Miller and C. Valasek, "Adventures in automotive networks and control units," *Defense*, vol. 21, no. 260-264, pp. 15–31, 2013.

[10] S. Nie, L. Liu, and Y. Du, "Free-fall: hacking tesla from wireless to can bus," *Briefing, Black Hat USA*, vol. 25, pp. 1–16, 2017.

[11] M. Yan, G. Harpak, and J. Li, *Security Research on mercedes-benz: From Hardware to Car Control*, vol. 28, pp. 1–38, Black Hat USA, Black Hat Briefing, USA, 2020.

[12] B. Groza, S. Murvay, A. Van Herrewege, and I. Verbauwhede, "Libra-can: a lightweight broadcast authentication protocol for controller area networks," in *Proceedings of the International Conference on Cryptology and Network Security*, pp. 185–200, Springer, Berlin, Germany, December 2012.

[13] S. Woo, H. J. Jo, and D. H. Lee, "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. 993–1006, 2014.

[14] 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*, vol. 17, no. 8, pp. 2248–2261, 2016.

[15] 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. 15, pp. 3107–3122, 2020.

[16] R. B. GmbH, *Can Specification*, Robert Bosch GmbH, Gerlingen, Germany, 2.0 edition, 1991.

[17] K. T. Cho and K. G. Shin, "Error handling of in-vehicle networks makes them vulnerable," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1044–1055, Vienna, Austria, October 2016.

[18] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.

[19] Y. Kim, A. Perrig, and G. Tsudik, "Simple and fault-tolerant key agreement for dynamic collaborative groups," in *Proceedings of the 7th ACM Conference on Computer and Communications Security*, pp. 235–244, Athens, Greece, November 2000.

[20] D. K. Nilsson, U. E. Larson, and E. Jonsson, "Efficient In-Vehicle Delayed Data Authentication Based on Compound Message Authentication Codes," in *Proceedings of the 2008 IEEE 68th Vehicular Technology Conference*, pp. 1–5, Calgary, Canada, September 2008.

[21] A. Van Herrewege, D. Singelee, and I. Verbauwhede, "Canauth-a simple, backward compatible broadcast authentication protocol for can bus," in *Proceedings of the ECRYPT Workshop on Lightweight Cryptography*, p. 20, Louvain-la-Neuve, Belgium, November 2011.

[22] A. Perrig, R. Canetti, D. Song, and J. D. Tygar, "Efficient and secure source authentication for multicast," in *Proceedings of the Network and Distributed System Security Symposium, NDSS*, vol. 2001, pp. 35–46, San Diego, California, USA, January 2001.

[23] 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.

[24] W. A. Farag, "Cantrack: Enhancing Automotive Can Bus Security Using Intuitive Encryption algorithms," in *Proceedings of the 7th International Conference on Modeling, Simulation, and Applied Optimization (ICMSAO)*, pp. 1–5, Sharjah, United Arab Emirates, UAE, April 2017.

[25] S. Fassak, Y. E. H. El Idrissi, N. Zahid, and M. Jedra, "A Secure Protocol for Session Keys Establishment between Ecus in the Can bus," in *Proceedings of the 2017 International Conference on Wireless Networks and Mobile Communications*, pp. 1–6, WINCOM), Morocco, Rabat, November 2017.

[26] Q. Pan and J. Tan, "A Dynamic Key Generation Scheme Based on Can bus," in *Proceedings of the 2019 10th International Conference on Information Technology in Medicine and Education (ITME)*, pp. 564–569, IEEE, Qingdao, China, August 2019.

[27] C. W. Lin, Q. Zhu, and A. Sangiovanni-Vincentelli, "Security-aware modeling and efficient mapping for can-based real-time distributed automotive systems," *IEEE Embedded Systems Letters*, vol. 7, no. 1, pp. 11–14, 2014.

[28] S. Jain and J. Guajardo, "Physical layer group key agreement for automotive controller area networks," in *Proceedings of the International Conference on Cryptographic Hardware and Embedded Systems*, pp. 85–105, Springer, Santa Barbara, CA, USA, Feb 2016.

[29] S. Nürnberger and C. Rossow, "- vatiCAN - vetted, authenticated CAN bus," in *Proceedings of the International Conference on Cryptographic Hardware and Embedded Systems*, pp. 106–124, Springer, Santa Barbara, CA, USA, August 2016.

[30] A. Humayed and B. Luo, "Using id-hopping to defend against targeted dos on can," in *Proceedings of the 1st International Workshop on Safe Control of Connected and Autonomous Vehicles*, pp. 19–26, New York NY, USA, April 2017.

[31] A. S. Siddiqui, Y. Gui, J. Plusquellic, and F. Saqib, "Secure communication over canbus," in *Proceedings of the 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 1264–1267, IEEE, Boston, MA, USA, August 2017.

[32] R. Canetti and H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels," in *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 453–474, Springer, Berlin, Heidelberg, April 2001.

[33] M. Ouaissa, M. Houmer, and M. Ouaissa, "An enhanced authentication protocol based group for vehicular communications over 5G networks," in *Proceedings of the 2020 3rd International Conference on Advanced Communication Technologies and Networking (CommNet)*, pp. 1–8, Marrakech, Morocco, September 2020.