




An Attribute-Isolated Secure Communication Architecture for Intelligent Connected Vehicles

Mu Han , Member, IEEE, Ailan Wan, Fengwei Zhang , and Shidian Ma 

Abstract—The rapidly increasing connectedness of modern vehicles leads to new security challenges for intelligent connected vehicles (ICVs), where some potential attackers can achieve unauthorized access to gain control of the vehicle by injecting malicious information into in-vehicle electronic control units (ECUs). Therefore, in this paper, a secure attribute-isolated communication architecture for an ICV, which introduces attributes into the ECUs to achieve authorized access among the ECU nodes is proposed. First, an analysis of the functional attributes of all of the in-vehicle ECUs in an intelligent connected environment and a division of the functional attributes of the ECUs into five classifications are performed. Second, based on the above-classified attributes, a secure attribute-isolated communication architecture is demonstrated. The ECUs have different access rights, allowing only the ECUs with the same functional attributes in the internal network of the vehicle to communicate. Then, it is proven that the proposed architecture can resist forgery and eavesdropping attacks under the random oracle model. Finally, the secure attribute-isolated communication architecture is constructed in a hardware environment and evaluated with an in-vehicle network simulator (IVNS). The evaluation results show that the average memory usage with 120 ECUs and 100 messages is below 40 MB and the bus load can be reduced to 18.96% using the proposed security architecture compared to the bus load of existing architectures. Therefore, the proposed secure attribute-isolated communication architecture solves the problem of the tradeoff between the security threat of unauthorized access and the high bus load of existing in-vehicle architectures.

Index Terms—ICV, in-vehicle network, security, attribute-isolated architecture, ECU functional attributes.

Manuscript received October 16, 2018; revised January 16, 2019, June 21, 2019, and January 30, 2020; accepted September 14, 2020. Date of publication September 29, 2020; date of current version November 23, 2020. This work was supported in part by the Six Talent Peaks Project of Jiangsu Province (DZXX-012), in part by Natural Science Fund for Colleges and Universities in Jiangsu Province (12KJD580002), in part by Jiangsu Graduate Innovation Fund (KYLX_1057), and in part by Key Research and Development Plan of Jiangsu province in 2017 (Industry Foresight and Generic Key Technology) (BE2017035). (Corresponding author: Mu Han.)

Mu Han is with the School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang 212013, China, and also with the COMPASS Lab, Wayne State University, Detroit, MI 48202 USA (e-mail: hanmu@ujs.edu.cn).

Ailan Wan is with the School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang 212013, China (e-mail: 1962130808@qq.com).

Fengwei Zhang was with the COMPASS Lab, Department of Computer Science, Wayne State University, Detroit, MI 48282 USA and is now with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518000, China (e-mail: zhangfw@sustech.edu.cn).

Shidian Ma is with the Automotive Engineering Research Institute, Jiangsu University, Zhenjiang 212013, China (e-mail: masd@ujs.edu.cn).

Color versions of one or more of the figures in this article are available online at <https://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TIV.2020.3027717

I. INTRODUCTION

RECENTLY, the rapid convergence of vehicles and information technology has resulted in a rapid increase in modern vehicles connected to the Internet [1]. Such a connection can make vehicles become rich sources of data, including both personal and vehicular information. However, it also means that vehicles inevitably become lucrative targets for hackers. In 2013, it was reported that a mass production hybrid vehicle had been cracked by hackers [2], who illegally manipulated the brake system through the on-board diagnostics (OBD) interface, allowing the hackers to cause traffic accidents and threaten the lives of the occupants. In 2015, the details of security attacks on a SUV have been unveiled [3]. The attackers were able to remotely invade the electronic control units (ECUs) through in-vehicle entertainment systems, and achieve remote control of the vehicle's speed, air conditioning and windshield wipers. Subsequently, in 2017, other researchers hacked some other cars. They showed that they could remotely control the vehicle including critical vehicle controls. They showed that they could remotely control the vehicle including critical vehicle controls [4].

The security loopholes mentioned above originated from the limitations of the traditional in-vehicle network architecture: 1) The traditional in-vehicle network architecture is a closed environment, that is insufficiently adapted to the open environment of modern intelligent connected vehicles (ICVs) [5]. Any devices connected to the vehicle can obtain access to the in-vehicle information via Wi-Fi, Bluetooth or OBD interfaces. This increase in interconnections expands the attack surface of the vehicle [6], [7]; 2) The communication framework of the in-vehicle network has broadcast characteristics, in which the ECUs (nodes) exchange information in the form of plaintext [8]. Each ECU can communicate with other ECUs without requiring source or destination addresses. Hence, an attacker who infiltrates an ECU can easily impersonate any other ECU and finally achieve remote control of the vehicle.

In this paper, an ECU access control mechanism for an ICV is designed, which achieves attribute-isolated communication among all of the ECUs. The ECUs' access control mechanism solves the security threat of unauthorized access. Additionally, it reduces the high bus load of existing in-vehicle architectures. The main contributions of this paper are as follows:

1) An analysis of the functional attributes of in-vehicle ECUs is performed. According to the impact of the passenger's functional requirements and the traffic environment on vehicles under intelligent connected environment, we divide the functional

attributes of ECUs into five classifications: perception, decision, control, execution, and service.

2) Based on the above classified ECU functional attributes, we propose an innovative in-vehicle secure attribute-isolated communication architecture. The architecture allows the ECUs with the same functional attributes in the internal network of the vehicle to communicate and isolates the ECUs with different functional attributes, achieving the purpose of access control and reducing the bus load. Then we prove that the proposed architecture can resist a forgery attack and an eavesdropping attack under the random oracle model.

3) We construct the attribute-isolated communication architecture in a hardware environment and evaluate the architecture with an in-vehicle network simulator (IVNS). The evaluation results indicate that the architecture is more efficient than existing schemes in terms of computation time, average storage consumption and bus load.

The rest of this paper is organized as follows: In Section II, we review more related works. In Section III, ECU functional attributes are classified. In Section IV, a novel attribute-isolated communication for the in-vehicle network is proposed. In Section V, we present a theoretical analysis of the security for the novel architecture. In Section VI, an evaluating experiment for the novel architecture is conducted and discussed. Finally, Section VII presents the conclusions and future work resulting from this study.

II. RELATED WORK

Researchers have been moving forward to design a secure in-vehicle network architecture to solve the information security problem of the in-vehicle network under the ICV environment. The first method to ensure in-vehicle network security was presented by Wolf *et al.* [9], who constructed a broadcast communication architecture based on the technology of encryption. In this architecture, all of the ECUs are connected to a gateway electronic control unit (GECU), and encrypted secret information is transmitted between the GECU and the others. Another approach to secure in-vehicle communication was proposed by Nilsson *et al.* [10], who first introduced message authentication codes (MACs) to authenticate the ECUs, overcoming the shortcoming that an ECU identity is easily impersonated. Nevertheless, the authentication process increases the load of the controller area network (CAN) bus, making the approach unsuitable for an in-vehicle real-time communication environment.

Groza *et al.* proposed a series of lightweight broadcast authentication communication solutions for an in-vehicle CAN such as EPSB (efficient protocols for a secure broadcast in controller area networks) and Libra-CAN (a lightweight broadcast authentication protocol for controller area networks) [11]–[13]. In their scheme, ECUs implement broadcast authentication protocols based on key-chains and time synchronization, meanwhile the limited data payload of the CAN data frame in the authentication process is considered. However, the total number of data frames in the vehicle network doubles at minimum when the data payload is used for MAC in these schemes, since it requires one data frame containing the original data and at least one data

frame containing the MAC. Hence, these schemes (including the full-length MAC) rapidly increase the load of the CAN bus and are not suitable for deployment in the vehicle environment. Jackson *et al.* went a step further, using a truncated MAC code (Mini-MAC code) to reduce the consumption of in-vehicle limited resource [14], but this approach weakened the security of the scheme and interactive information among the ECUs can be leaked easily.

In 2012, Robert Bosch GmbH developed a new communication protocol [15], known as CAN with flexible data rate (CAN-FD) to solve the problem of the existing security architectures are inapplicable of directly assisting in-vehicle CAN because of the limited data payload [16]. The CAN-FD design is based on CAN, with the following advantages: First, it has a higher bandwidth and a larger data payload. Second, its physical layer and topologies can be maintained. Soon afterward, Samuel *et al.* proposed a practical security architecture (PSAC) for in-vehicle CAN-FD [17], [18]. In PSAC, ECUs derive session keys with a GECU in a fixed order and perform authentication and encryption based on the Keyed-Hash MAC and advanced encryption standard (AES). Patsakis *et al.* proposed a distributed secure in-vehicle communication architecture (DSCA) for modern vehicles under a CAN-FD [19]. In the DSCA, the ECUs participate in a secure multi-party computation scheme to perform authentication and encryption. However, all of the ECUs need to perform decryption, which rapidly increases the bus load, limiting the applicability of this approach in real-time vehicle systems.

The above architectures under a CAN-FD do not fully consider the access control mechanism, and unauthorized attackers can also receive the in-vehicle private information. Meanwhile, the bus load of these architectures is high. In this paper, we propose an isolated architecture based on ECU functional attributes under CAN-FD. The proposed architecture not only has an access structure but also can reduce the bus load when compared with [18], [19].

III. ECU FUNCTIONAL ATTRIBUTE CLASSIFICATION

A. Attribute Clustering

The traditional vehicle is a typical driver-centered system. As shown in Fig. 1, a driver perceives changes of the traffic environment through visual and auditory senses. Meanwhile, the driver judges the current environment through their brain and makes driving decisions to control the movement of their hands and feet, completing the manipulation of the vehicle.

With the rapid development of artificial intelligence, internet technology, communication technology and computer technology, ICV based on electrification, intellectualization and networking has become a significant trend in the automotive industry. An ICV is mainly embodied by the replacement of manual operation with automatic driving, which can compensate for the shortcomings of the human sensory ability and reduce the driving manipulation intensity. The behaviour and running state of the vehicle are controllable and predictable. Therefore, traffic accidents caused by human factors can be eliminated, and travel paths can be planned according to real-time road condition

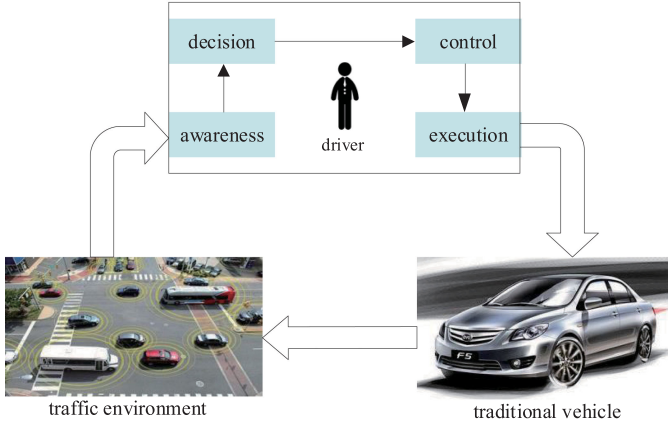


Fig. 1. The manipulation of the traditional vehicle [20].

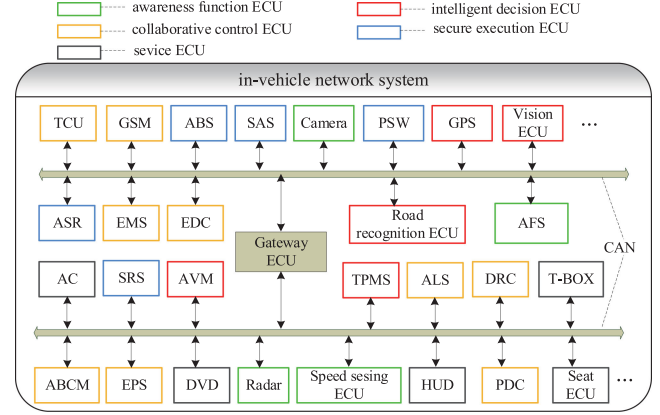


Fig. 3. In-vehicle network system.

TABLE I
ATTRIBUTE CLUSTERING OF ECUS

ECU functional attributes	ECU Functional attributes [25], [26], [27]
Att _P : Perception function attribute	<ul style="list-style-type: none"> Obtain both the in-vehicle ECUs' states and the data from V2X. Camera, radar, speed sensing ECU, etc.
Att _I : Intelligent decision function attribute	<ul style="list-style-type: none"> Analyze all of the correlative relationships between the ECUs of the whole vehicle. Vision ECU, around view monitor (AVM), Road recognition ECU, etc.
Att _C : Collaborative control function attribute	<ul style="list-style-type: none"> Achieve coordinated control and all around supervision of the whole vehicle Transmission Control Unit (TCU), Engine Management System (EMS), Electronic Stability Program (ESP), etc.
Att _{S1} : Secure execution function attribute	<ul style="list-style-type: none"> Ensure vehicle and personal safe. Antilock Brake System (ABS), Traction Control System (TCS), Safety Assistance System (SAS), etc.
Att _{S2} : Service function attribute	<ul style="list-style-type: none"> Ensure passenger comfortable. T-BOX, Air Conditioner (AC), DVD, etc.

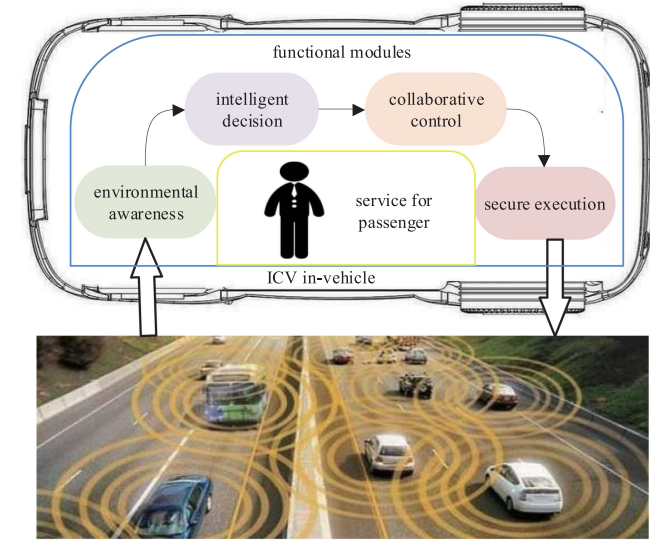


Fig. 2. The manipulation of the ICV [23].

information. Ultimately, zero casualties and zero congestion in the road transport can be achieved [21].

As shown in Fig. 2, an ICV integrates multiple information and physical function modules such as an environmental awareness module, intelligent decision module, collaborative control module, secure execution module and service module. The ICV can realize safe, comfortable, energy-saving, and efficient driving, and can eventually replace a new generation of vehicles operated by human beings [22].

These functional modules of the ICV are made possible by a range of 50 to 70 in-vehicle computers networked together, called electronic control units (ECUs) [24]. The ECUs exchange information with remote access equipment through wireless communication to sense traffic information. Meanwhile, the ECUs transmit operation data and control instructions through the in-vehicle network (CAN), as shown in Fig. 3.

Based on an analysis of the manipulation of the traditional vehicle, the manipulation of the ICV and the in-vehicle network

system, we classify the functional attributes of ECUs into five functional attributes: Att_P, Att_I, Att_C, Att_{S1} and Att_{S2}, as shown in Table I.

B. The Scalability of Attribute Clustering

Based on the above-classified ECU functional attributes, an innovative in-vehicle attribute-isolated broadcast communication architecture is constructed. We will show that the attribute clustering of ECUs is scalable for the novel architecture.

1) The scalability of the ECU function attributes. The traditional vehicle is a typical driver-centered system, which perceives changes of the traffic environment, judges the current environment and forms driving decisions, completing the manipulation of the traditional vehicle. Compared with the traditional vehicle, an ICV is mainly embodied by replacement of manual operation. In an ICV system, there are 50 to 70 ECUs networked together, which like the human brain, hands, eyes and feet, achieve environmental awareness, intelligent decision-making, collaborative control, secure execution, and service. If a new ECU is added to the ICV, its functional attribute of it should be in the above classified functional attributes.

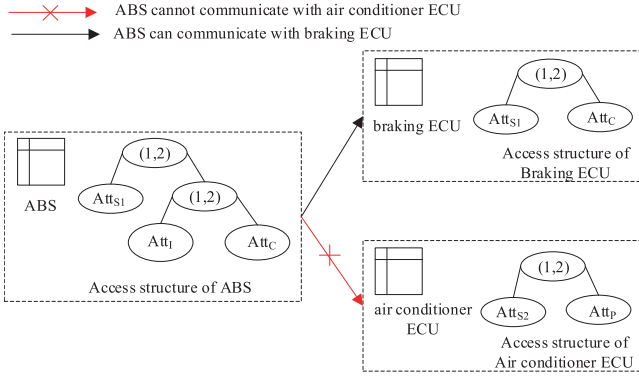


Fig. 4. The isolated communication based on ABS access structure.

2) Attribute communication is scalable, that is, ECUs with any one of the same attributes can perform isolated-communication. For example, ABS has three functional attributes $\{Att_{S1}, Att_I, Att_C\}$, where Att_{S1} is the main functional attribute of the ABS. Setting the access structure for the ABS based on its functional attributes, and allowing other ECUs with one of the above three functional attributes to exchange information. Fig. 4 is a diagram of the isolated communication based on the ABS access structure, which shows that leaf nodes are composed of ECU functional attributes and each non-leaf node consists of a pair of threshold values (1, 2). Hence, the other ECUs with functional attributes $\{Att_{S1}\}$, $\{Att_I\}$, $\{Att_C\}$, $\{Att_{S1}, Att_I\}$, $\{Att_{S1}, Att_C\}$, $\{Att_I, Att_C\}$, $\{Att_{S1}, Att_I, Att_C\}$ can exchange information with the ABS, e.g., the braking ECU with functional attributes $\{Att_{S1}, Att_C\}$ can communicate with the ABS. However, the air conditioner ECU with different functional attributes $\{Att_P, Att_{S2}\}$ cannot communicate with the ABS.

IV. IN-VEHICLE ATTRIBUTE-ISOLATED COMMUNICATION ARCHITECTURE

In this section, based on the above ECU attribute clustering, the proposed in-vehicle attribute-isolated communication architecture is elaborated. The novel architecture consists of a gateway electronic control unit (GECU) and electronic control units (ECUs) which are equipped in vehicles. The specific functions are as follows:

GECU: The GECU¹ functions as the trust authority [28] and verifies the identity of the ECUs. Meanwhile, the GECU has sufficient computation power and storage capacity, typically well above those of a general ECU.

ECU: Before being allowed to interact with information, the ECUs must register with the GECU. To function as a sender ECU, it needs to set an access structure for the receiver ECUs based on its functional attributes. Two ECUs can only communicate when the functional attributes of a receiver ECU satisfy the access structure.

The proposed in-vehicle attribute-isolated communication architecture consists of five phases, namely “system initialization”, “registration”, “setting the access structure”,

¹GECU is the trusted third party and is free from security leakages.

TABLE II
NOTATIONS AND DESCRIPTIONS

Notations	Descriptions
$GECU$	Gateway Electronic Control Units (Assuming that the GECU is trusted)
ECU	Electronic control unit
ECU_I	The I -th ECU
$SK_{CP-ABE}^{ECU_I}$	Attribute private key of ECU_I
ID_{ECU_I}	Identity of ECU_I
Att_i	Functional attribute of ECU_I
S_i	Functional attribute set of ECU_I
Sig_I	The signature of ECU_I
CT_{ECU_I}	The ciphertext of ECU_I
CTR_I	The counter of ECU_I
ACP_I	The access structure of ECU_I

“attribute-isolated communication” and “updating the ciphertext and attribute private key”.

P1. System initialization. The GECU publishes the public parameters and generates the master key.

P2. Registration. The ECUs register their identities information to the GECU.

P3. Setting the matching strategy. We set the matching strategy for the access structure of the ciphertext and the attribute private key.

P4. Attribute-isolated communication. The ECUs perform attribute-isolated communication based on the matching strategy.

P5. Updating the ciphertext and attribute private key. This phase prevents attackers from obtaining in-vehicle private data.

In the following, we present the details of each phase. The notations used in the five phases are listed in Table II.

A. System Initialization

Step 1: The GECU inputs the secure parameter k and generates two additive groups G_0 and a multiplicative group G_1 . Define a bilinear mapping $e : G_0 \times G_0 \rightarrow G_1$, and two generators p_1, p_2 of G_0 and G_1 , respectively, where G_0 and G_1 have prime order q .

Step 2: The GECU randomly chooses $\alpha, \beta, \theta \in Z_q^*$ and a hash function $H : \{0, 1\}^* \rightarrow Z_q^*$.

Step 3: The GECU publishes the public parameters: $\{G_0, G_1, e, H, p_1, p_2, \theta p_2, \theta^2 p_2, Y = e(p_1, p_2)^{\alpha(\beta-1)}, e(p_1, p_2)^{\alpha\beta}\}$. Meanwhile, the public key is $PK_{GECU} = SK_{GECU} P_2$, and the master key is $MK = SK_{GECU} = \alpha p_1$.

B. Registration

After completing the system initialization, the GECU performs the registration phase to verify the ECU identity, preventing attackers from impersonating the ECU identity. Algorithm 1 indicates the registration process. The concrete steps are as follows.

Step 1: ECU_1 sends registration request information to the GECU. The registration information is generated as follows.

1) ECU_I chooses (PK_{ECU_I}, SK_{ECU_I}) as its public and private key pairs, where $SK_{ECU_I} = r_1 (r_1 \in Z_q^*)$, $PK_{ECU_I} = SK_{ECU_I} p_2$.

2) ECU_I computes $H(ID_{ECU_I})$ by its identity ID_{ECU_I} and selects a secret random number r_2 ($r_2 \in Z_q^*$) to generate the request information (V_{ECU_I}, W_{ECU_I}) where $(V_{ECU_I} = PK_{ECU_I} H(ID_{ECU_I})), W_{ECU_I} = r_2 p_2$.

3) ECU_I signs the request information (V_{ECU_I}, W_{ECU_I}) through SK_{ECU_I} to obtain the signature information $Sig_I = sig_{SK_{ECU_I}}(V_{ECU_I} || W_{ECU_I})$.

4) ECU_I computes $H(ID_{ECU_I})$ and uses PK_{GECU} to encrypt V_{ECU_I}, W_{ECU_I} and $H(ID_{ECU_I})$. ECU_I sends $Msg_1(E_{PK_{GECU}}(V_{ECU_I} || W_{ECU_I} || H(ID_{ECU_I})) || Sig_I || t)$ to GECU.

Step 2: After receiving Msg_1 , the GECU takes the following actions to verify the legitimacy of ECU_I .

1) The GECU verifies the validity of the timestamp by formula (1), where t' is the current time. T is the maximum time difference allowed for the vehicle.

$$(\Delta t = t' - t) < T \quad (1)$$

2) Once verified successfully, the GECU decrypts $E_{PK_{GECU}}(V_{ECU_I} || W_{ECU_I} || H(ID_{ECU_I}))$ in Msg_1 by SK_{GECU} and obtains V_{ECU_I}, W_{ECU_I} and $H(ID_{ECU_I})$.

3) The GECU confirms the correctness of the request information by verifying Sig_I . The GECU uses formula (2) to verify Sig_I . If $Ver_{PK_{ECU_I}}(V_{ECU_I} || W_{ECU_I}, Sig_I) = Ver_{PK_{ECU_I}}(V_{ECU_I} || W_{ECU_I}, Sig_{SK_{ECU_I}}(V_{ECU_I} || W_{ECU_I})) = true$, it indicates that Msg_1 has not been forged. Otherwise, the GECU discards Msg_1 .

$$Ver_{PK_{ECU_I}}(x, y) = \begin{cases} true & y = sig_{SK_{ECU_I}}(x) \\ false & y \neq sig_{SK_{ECU_I}}(x) \end{cases} \quad (2)$$

where x is the request information (V_{ECU_I}, W_{ECU_I}) and y is the signature information $Sig_I = sig_{SK_{ECU_I}}(V_{ECU_I} || W_{ECU_I})$.

4) GECU randomly selects $r_3 \in Z_q^*$ and computes $R = W_{ECU_I} + r_3 p_2$, $L = r_3 + SK_{GECU} V_{ECU_I}$. GECU verifies the legal identity of ECU_I by formula (3). If formula (3) is established, it indicates that the identity of ECU_I is legal. Otherwise, ECU_I is forged.

$$LP + W_{ECU_I} = R + PK_{GECU} V_{ECU_I} \quad (3)$$

The process of the verification is shown in formula (4). If formula (4) is established, it indicates that the identity of ECU_I is legal. Otherwise, ECU_I is forged, and the GECU refuses the request information and terminates the session.

$$\begin{aligned} LP + W_{ECU_I} &= (r_3 + SK_{GECU} V_{ECU_I}) p_2 + r_2 p_2 \\ &= (r_3 + SK_{GECU} V_{ECU_I} + r_2) p_2 \\ &= (r_2 + r_3) p_2 + SK_{GECU} V_{ECU_I} p_2 \\ &= R + PK_{GECU} V_{ECU_I} \end{aligned} \quad (4)$$

5) After verifying the legal identity of ECU_I , the GECU stores the set of legal ECUs and returns the successful registration information $Msg_2(E_{PK_{GECU}}(H(ID_{ECU_I})) || t')$ to the ECU.

Step 3: ECU_I decrypts $E_{PK_{GECU}}(H(ID_{ECU_I}))$ in Msg_2 through SK_{ECU_I} . This indicates that ECU_I successfully registers in the GECU.

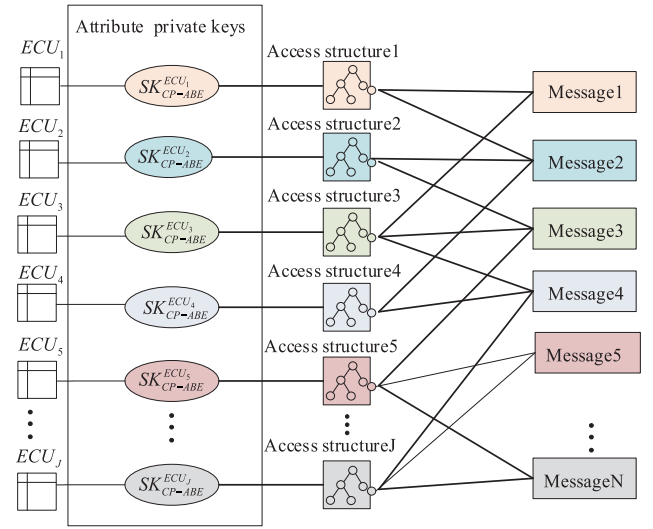


Fig. 5. Matching strategy for the access structure and attribute private key.

Algorithm 1: ECU Registration Protocol (ECU_REGISTRATION).

1: ECU_I : Generate the registration request information

2: $ECU_I \rightarrow GECU$:

$Msg_1(E_{PK_{GECU}}(V_{ECU_I} || W_{ECU_I} || H(ID_{ECU_I})) || Sig_I || t)$

where $V_{ECU_I} = PK_{ECU_I} H(ID_{ECU_I}), W_{ECU_I} = r_2 p_2$

3: GECU: Verify the legitimate identity of ECU_I

4: if $(\Delta t = t' - t) < T$ is valid then

GECU: Decrypt $E_{PK_{GECU}}(V_{ECU_I} || W_{ECU_I} || H(ID_{ECU_I}))$ in Msg_1 by SK_{GECU} and verify Sig_I

else

GECU: Refuse the request information

endif

5: if $Ver_{PK_{ECU_I}}(Sig_I, V_{ECU_I} || W_{ECU_I}) = true$, then

GECU: Compute $R = W_{ECU_I} + r_3 p_2$,

$L = r_3 + SK_{GECU} V_{ECU_I}$

else

GECU: Refuse the request information

6: if $LP_2 + W_{ECU_I} = R + PK_{GECU} V_{ECU_I}$ then GECU: The legal identity of the ECU is successfully authenticated, and then $GECU \rightarrow ECU_I$:

$Msg_2(E_{PK_{GECU}}(H(ID_{ECU_I})) || t')$

endif

7: ECU_I : Successfully registered

endif

C. Setting the Matching Strategy

After the ECUs successfully register in the GECU, we set the matching strategy for the access structure of the ciphertext and the attribute private key. As shown in Fig. 5, the in-vehicle ECUs are logically separated from their access rights. The ECUs are described by functional attributes and obtain attribute private keys according to their functional attributes. The ECUs can communicate only when their attribute private keys can

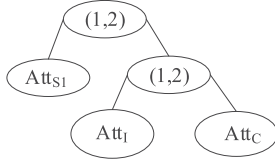


Fig. 6. Access structure of ABS.

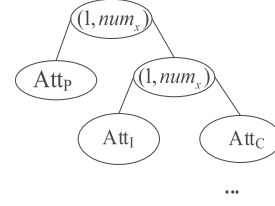


Fig. 8. Access Structure.

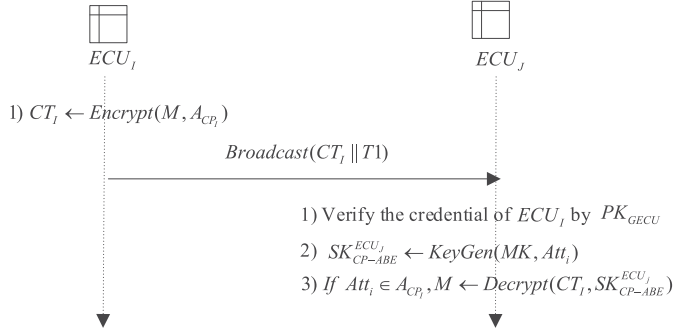


Fig. 7. In-vehicle attribute-isolated communication.

decrypt the ciphertext. Therefore, the matching strategy realizes the isolated communication among the ECUs with the same functional attributes.

For example, the antilock brake system (ABS), which can quickly judge the lock state of a wheel according to the speed signal from each wheel speed ECU to ensure the vehicle safety, has three functional attributes: Att_I : the perception function attribute, Att_C : the collaborative control function attribute, and Att_{S1} : the secure execution function attribute. We set the access structure for ABS based on its functional attributes, as shown in Fig. 6. In addition, the ABS generates the attribute private key based on its functional attributes. Therefore, the attribute private key matches the access structure based on the ABS functional attributes.

D. Attribute-Isolated Communication

Based on the above matching strategy and ciphertext-policy attribute-based encryption algorithm (CP-ABE), the ECUs perform attribute-isolated communication, thereby allowing only the authorized ECUs to obtain the in-vehicle private information. The specific design of our scheme is shown in Fig. 7. We take ECU_I and ECU_J to show how the ECUs can communicate, ECU_I and ECU_J have the same functional attributes Att_i . The concrete steps are as follows.

1) ECU_I Generates the Ciphertext: ECU_I performs the following steps to broadcast the ciphertext:

i) ECU_I sets the access structure A_{CP_I} as shown in Fig. 1. Every non-leaf node of the tree represents a threshold k_r and the number of a children node num_x ($1 \leq num_x \leq 5$). Each leaf node r of the tree is described by its functional attributes.

ii) ECU_I chooses $s \in Z_q^*$ and a polynomial q_r of degree $d_r = k_r - 1$ in A_{CP_I} , where k_r is the threshold of the node r in A_{CP_I} , $q_r(0) = s$. Let N_r denote the set of all leaf nodes in A_{CP_I} .

ECU_I computes the ciphertext by formula (6) as follows:

$$CT_{IJ} = (A_{CP_I}, \tilde{C}, C, C', \forall r \in N_r : C'_r, C''_r) \quad (5)$$

Where $\tilde{C} = e(p_1, p_2)^{\alpha \beta s} M$, $C = sp_1$, $C' = Y^s$, $C'_r = q_r(0)H(Att_i)p_2 + q_r(0)\theta^2 p_2$, and $C''_r = q_r(0)\theta p_2$.

Meanwhile, ECU_I broadcasts $M_{sg_{B1}}(CT_I || T1)$ in the vehicle.

2) ECU_J Obtains M From the Ciphertext: ECU_J performs the following steps to achieve M from the broadcast ciphertext.

i) ECU_J takes Att_i, MK as an input, where Att_i is the functional attribute of ECU_I . ECU_J randomly selects $k_j, l_j \in Z_q^*$ for each attribute $Att_i \in S_j$ and then computes $h_j = H(Att_i)$. Subsequently, ECU_J computes the attribute private key $SK_{CP-ABE}^{ECU_J}$ as: $SK_{CP-ABE}^{ECU_J} = (D_j, \forall Att_i \in S_j : D'_j, D''_j)$ where $D_j = \alpha p_2 + l_j k_j p_2$, $D'_j = k_j p_1 + h_j p_1$, $D''_j = l_j k_j \theta p_1$.

ii) ECU_J uses the recursive algorithm $DecryptNode(CT_I, SK_{CP-ABE}^{ECU_J}, r)$ and inputs $CT_I, SK_{CP-ABE}^{ECU_J}$ and node r of the access structure A_{CP_I} . $SK_{CP-ABE}^{ECU_J}$ is related to the attribute set S_j and the attribute Att_i . The definition of $DecryptNode(CT_I, SK_{CP-ABE}^{ECU_J}, r)$ is as follows:

$$\begin{aligned} DecryptNode(CT_I, SK_{CP-ABE}^{ECU_J}, r) &= \frac{e(D'_j, C'_r)}{e(D''_j, C''_r)} \\ &= e(l_j k_j p_1, q_r(0) p_2) \\ &= e(p_1, p_2)^{l_j k_j q_r(0)} \end{aligned} \quad (6)$$

We denote:

$$B = DecryptNode(CT_I, SK_{CP-ABE}^{ECU_J}, r) = e(p_1, p_2)^{l_j k_j s}, \text{ where } q_r(0) = s.$$

After performing $DecryptNode(CT_I, SK_{CP-ABE}^{ECU_J}, r)$, ECU_J can obtain M as follows:

$$\begin{aligned} Decrypt(CT_I, SK_{CP-ABE}^{ECU_J}) &= \frac{B \cdot \tilde{C}}{e(C, D_j) \cdot C'} \\ &= \frac{e(p_1, p_2)^{l_j k_j s} \cdot e(p_1, p_2)^{\alpha \beta s} M}{e(p_1, p_2)^{s(\alpha + l_j k_j)} \cdot e(p_1, p_2)^{\alpha(\beta - 1)s}} \\ &= M \cdot e(p_1, p_2)^{s(\alpha - \alpha)} \\ &= M \end{aligned} \quad (7)$$

E. Updating the Ciphertext and Attribute Private Key

The proposed attribute-isolated communication architecture introduces a counter mechanism in the traditional CP-ABE

scheme to update the ciphertext and attribute private key of each ECU. We specify that the counter is integrated into the access structure when the sender ECU generates the ciphertext each time, hence the receiver ECUs must synchronize the counters and update their own attribute private key. This effectively prevents attackers from obtaining in-vehicle private data. The concrete steps are as follows.

1) *ECU_I Updates the Ciphertext*: When a vehicle stalls, any broadcast interaction ends, and *ECU_I* will invalidate its previous ciphertext *CT_I*. The next time the vehicle is started, *ECU_I* must generate a new ciphertext as follows.

i) *ECU_I* manages its own counter value *CTR_I* and computes $H(Att_i \oplus CTR_I)$, and then generates the new ciphertext *CT'_I* by formula (8).

$$CT'_I = A_{CP_I}, \tilde{C}, C, C', \forall r \in N_r : C'_{r_{new}}, C''_r \quad (8)$$

where $\tilde{C} = e(p_1, p_2)^{\alpha\beta s} M$, $C = sp_1$, $C' = Y^s$, $C'_{r_{new}} = q_r(0)H(Att_i \oplus CTR_I)p_2 + q_r(0)\theta^2 p_2$, and $C''_r = q_r(0)\theta p_2$.

ii) *ECU_I* broadcasts the new ciphertext *CT'_I* in the vehicle and increments *CTR_I*.

2) *ECU_J Updates the Attribute Private Key*: After receiving the ciphertext, *ECU_J* manages the counter of *ECU_J* and computes $h'_j = H(Att_i \oplus CTR_I)$, and then generates the new attribute private key $SK_{CP-ABE}^{ECU_J}$ as $SK_{CP-ABE}^{ECU_J} = (D_j, \forall Att_i \in S_j : D'_{ik_{new}}, D''_{ik})$ where $D_{ik} = \alpha p_2 + l_j k_j p_2$, $D'_{ik_{new}} = k_j p_1 + h'_j p_1$, and $D''_j = l_j k_j \theta p_1$. *ECU_J* increments the counter *CTR_I* of *ECU_I*.

V. SECURITY ANALYSIS OF THE PROPOSED SCHEME

In this section, we theoretically prove that the proposed architecture can resist forgery attack and eavesdropping attack under the random oracle model.

Theorem 1: Assuming that the Discrete Logarithm (DL) assumption is established, the ECU ID in the proposed attribute-isolated architecture can resist a forgery attack.

Proof: Assume that the attacker *A* can fake the real identity information *ID_{ECU_I}* of the legal ECU and generate a valid message *M_{sg₁}*, that is, *A* can calculate the effective value $V_{ECU_I} = PK_{ECU_I} H(ID_{ECU_I})$ of *M_{sg₁}*. The advantage of *A* attack success is *Adv_A*. We use *A* to construct an algorithm *A_{DL}* to solve the DL problem. ■

A_{DL} randomly chooses $\theta \in Z_q^*$, publishes the public parameters: $\{G_0, G_1, e, H, p_1, p_2, \theta p_2, \theta^2 p_2, Y = e(p_1, p_2)^{a(b-1)}, e(p_1, p_2)^{ab}\}$ and saves the master key $MK = ap_2$ secretly. *A* can make queries about *A_{DL}* to *q_{DL}* times.

Query: *A* makes queries about *ID_{ECU_I}*, and algorithm *A_{DL}* returns $V_{ECU_I} = PK_{ECU_I} H(ID_{ECU_I})$ to *A*.

Challenge: After *A* receives V_{ECU_I} , *A* uses (PK_{ECU_I}, V_{ECU_I}) to call algorithm *A_{DL}* and lets $H(ID_{ECU_I}) = a$. That is given PK_{ECU_I} , aPK_{ECU_I} , compute *a*.

The advantage of *A* challenge success in this process is $Adv_A = q_{DL} \cdot Adv_{DL}$. As can be seen from the difficult problems described in the preliminaries, the advantage *Adv_A* of the algorithm in successfully solving the DL problem in the polynomial time is negligible, hence the attacker *A* cannot counterfeit the ECU ID.

Theorem 2: Provided that the DL assumption is established, the ECU attributes in the proposed attribute-isolated architecture cannot be obtained.

Proof: Assume that the attacker *A* can obtain the real attributes *Att_i* of the ECU, that is, *A* can calculate the effective value $H(Att_i)p_1$ in the ECU attribute private key $SK_{CP-ABE}^{ECU_J}$. The advantage of *A* successful attack is *Adv_A*. We use *A* to construct algorithm *A_{DL}* to solve DL problem. ■

A_{DL} randomly chooses $\theta \in Z_q^*$, publishes the public parameters: $\{G_0, G_1, e, H, p_1, p_2, \theta p_2, \theta^2 p_2, Y = e(p_1, p_2)^{a(b-1)}, e(p_1, p_2)^{ab}\}$ and saves the master key $MK = ap_2$ secretly. *A* can make queries about *A_{DL}* to *q_{DL}* times.

Query: *A* makes queries about the real attributes *Att_i* of the ECU, and *A_{DL}* returns $H(Att_i)p_1$ to *A*.

Challenge: After *A* receives $H(Att_i)p_1$, *A* uses $(p_1, H(Att_i)p_1)$ to call algorithm *A_{DL}* and lets $H(Att_i) = a$. That is given p_1, ap_1 , compute *a*. The advantage of *A* challenge success in this process is $Adv_A = q_{DL} \cdot Adv_{DL}$. The advantage *Adv_A* of the algorithm in successfully solving the DL problem in the polynomial time is negligible. Hence the attacker *A* cannot obtain the ECU attributes and generate the ECU attribute private key.

Theorem 3: Assuming the DBDH (Determine Bilinear Diffie-Hellman) problem is difficult, then the scheme designed in this paper is CCA (Chosen-Ciphertext Attack) secure.

Proof: Challenger *C* randomly chooses $\theta \in Z_q^*$, publishes the public parameters: $\{G_0, G_1, e, H, p_1, p_2, \theta p_2, \theta^2 p_2, Y = e(p_1, p_2)^{a(b-1)}, e(p_1, p_2)^{ab}\}$ and saves the master key $MK = ap_2$. ■

Phase 1: *C* generates the attribute private key according to the attribute set *S_i* of the ECU: $SK_{CP-ABE}^{ECU_I}(D_i)$. Then *C* generates *CT* and broadcasts it according to formula (5) in the attribute-isolated communication process: $CT = A_{CP}, \tilde{C}, C, C', C'_r, C''_r$.

The attacker *A* intercepts ciphertext *CT* from the broadcast messages by eavesdropping and sends *CT* to *C*. *C* uses the decryption algorithm to obtain *M* and transmits it to *A*.

Challenge: *A* chooses two equal messages *M₀*, *M₁* and access structure A_{CP}^* . *A* sends (M_0, M_1, A_{CP}^*) to *C*. *C* sets *r* to be the set of the root node in A_{CP}^* after receiving (M_0, M_1, A_{CP}^*) . Then *C* randomly selects $b' \in \{0, 1\}$ and computes:

$$CT^* = (A_{CP}^*, \tilde{C} = Z \cdot M_{b'},$$

$$C = cp_1, C' = \frac{Z}{e(ap_1, cp_2)},$$

$$\forall r \in N_r :$$

$$C'_r = q_r(0)H(att_i)p_2 + q_r(0)\theta^2 p_2,$$

$$C''_r = q_r(0)\theta p_2)$$

Finally, *C* returns *CT** to *A*.

Phase 2: All of the queries in Phase 1 can be performed during Phase 2. However, if *A* asks for the decryption algorithm in the attribute-isolated communication phase, *C* will abort the simulation.

Guess: A outputs $b'' \in \{0, 1\}$ as a guess for b' . If $b'' = b'$, A will win the game. Otherwise, A fails. If the attacker A can win the CCA game in polynomial time with non-negligible advantage ε , then C can solve the DBDH enlarge 20% problem with non-negligible advantage ε' , where $\varepsilon' \geq \frac{1}{2}(\varepsilon - \delta)$ and δ is a negligible advantage. The probability analysis is as follows.

If $Z = e(p_1, p_2)^{abc}$, we can achieve that

$\Pr[A(p_1, p_2, ap_1, bp_1, cp_1, ap_2, bp_2, cp_2, e(p_1, p_2)^{abc}) = 1] = \Pr[b'' = b']$ where $|\Pr[b'' = b'] - \frac{1}{2}| \geq \varepsilon$. Otherwise, if Z is randomly chosen from G_1 , then

$\Pr[A(p_1, p_2, ap_1, bp_1, cp_1, ap_2, bp_2, cp_2, Z) = 1] = \Pr[b'' = b']$

Where $|\Pr[b'' = b'] - \frac{1}{2}| \leq \delta$ and δ is the advantage of breaking the semantic security and can be ignored. Hence, we can present that

$$\begin{aligned} & |\Pr[A(p_1, p_2, ap_1, bp_1, cp_1, ap_2, bp_2, cp_2, e(p_1, p_2)^{abc}) = 1] \\ & - \Pr[A(p_1, p_2, ap_1, bp_1, cp_1, ap_2, bp_2, cp_2, Z) = 1]| \\ & \geq \varepsilon - \delta \end{aligned}$$

Therefore, we can conclude that the proposed scheme satisfies CCA secure.

VI. SIMULATION AND EVALUATION

In this section, we first constructed the attribute isolated architecture using STMicroelectronics's automotive microcontrollers to extract the hardware parameters and then evaluated the architecture using the software In-Vehicle Network Simulator (IVNS). We compared the proposed architecture with the architecture suggested in [18] and [19] in terms of the computation time, average storage consumption, and bus load.

A. Simulation

1) *The Construction of the Attribute Isolated Architecture in the Hardware Environment:* To ensure that the software simulation follows reality as closely as possible, the simulation can be parametrized with measurements from real hardware. In the hardware-constructed experiment of the attribute isolated architecture, we used 12 STM32 microcontrollers with 12 address numbers of 0X0446 to 0X0451 as ECU nodes, and the 0X0449 address number node was used as the GECU. We first transplanted the portable system contiki in the integrated environment of keil's MDK5 and compiled the codes used in the experiment. As shown in Fig. 9, we built an attribute isolated architecture. The time parameters of the hardware were exported through the serial port. The specifications of the tools used in the hardware experiment and software simulation are shown in Table III.

i) *Extraction of the registration time parameters:* We measured the registration time of 12 ECUs in the attribute-isolated architecture respectively, as shown in Table IV. The average registration time for an ECU to perform one registration is approximately 0.36 ms. The average time that the GECU verifies one ECU is approximately 0.54 ms.

ii) *Extraction of the encryption and decryption time parameters:* The average execution time for encryption, attribute private key generation and decryption were measured by implementing

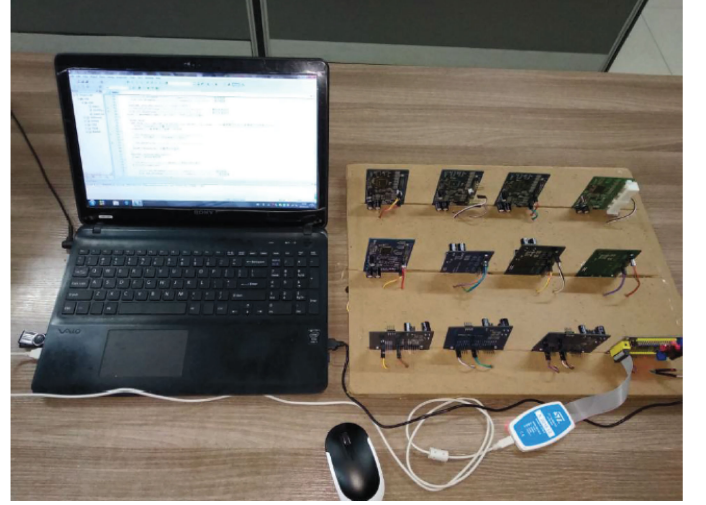


Fig. 9. Construction environment of the attribute-isolated architecture.

TABLE III
TOOLS USED FOR THE SIMULATION

Tools	Remarks
O/S	Contiki, Ubuntu Linux kernel-4.15.0-33-generic
Hardware	STM32
Compiler	MDK5
Hterm	Serial debugging tool
Software	IVNS
CPU	Intel core i3-370M 1.6 GHz
RAM	8GB
PC	Used to install these software packages

TABLE IV
REGISTRATION TIME EXTRACTED FROM THE HARDWARE

Node	ID	Registration time
GECU	0x0449	5.9 ms
ECU_1	0x0446	0.34 ms
ECU_2	0x0448	0.35 ms
ECU_3	0x044A	0.38 ms
ECU_4	0x0447	0.36 ms
ECU_5	0x044B	0.35 ms
ECU_6	0x044C	0.39 ms
ECU_7	0x044D	0.33 ms
ECU_8	0x044E	0.34 ms
ECU_9	0x044F	0.37 ms
ECU_10	0x0450	0.35 ms
ECU_11	0x0451	0.36 ms

the CP-ABE algorithms on 11 ECUs. We measured the average key generation time, the average encryption time, successful decryption time and failed decryption time from 1 byte to 100 bytes, as shown in Fig. 10. The average time for an ECU to generate the attribute private key is approximately 3 ms. The average encryption time on hardware is 4.8 ms. The average time for an ECU to fail to achieve decryption is 1.6 ms. The average time for an ECU to successfully achieve decryption is 7.5 ms.

2) *IVNS Simulation:* Then, to verify the performance of the proposed attribute isolated communication among the ECUs, the time parameters extracted from the hardware experiment are imported into a python database. We built a 64-bit system environment based on Ubuntu under the PC, with 1.6 GHz CPU

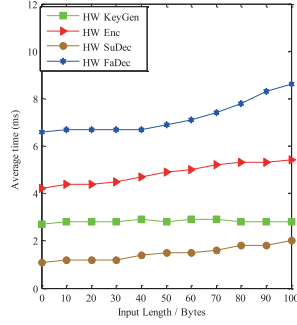


Fig. 10. CP-ABE performance for hardware implementation.

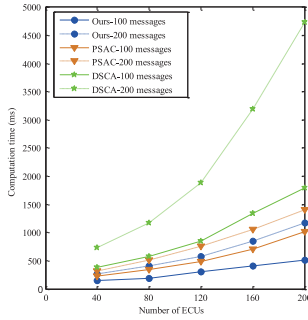


Fig. 11. Computation time.

and 8 GB RAM. Then, we built a simulator environment based on In-Vehicle Network Simulator (IVNS) which is developed by Artur Mrowca *et al.* [29]. To export the communication performance results, we created a monitor tag in the communication layer of the IVNS. The monitor tag outputs the simulation results into CSV files. The analysis of the simulation results is shown in the next subsection.

B. Evaluation and Comparison

In this subsection, we compare the proposed architecture with the architecture PSAC suggested in [18] and the architecture DSCA in [19] in terms of the computation time, average storage consumption, and bus load. For the convenience of the description, our proposed architecture is denoted as Ours.

1) *Computation Time*: We measure the computation time from the time that the ECUs process a message and transmits the encrypted messages in our scheme. The simulation result is shown in Fig. 11, if ECUs are added to the system for all of the architectures, the measured computation time needed to execute the simulation increases with an increasing number of ECUs. In addition, the DSCA performs the slowest, while the computation time for the PSAC with 100 messages is nearly equal to that of our architecture with 200 messages. Furthermore, the slope of the curves is higher when ECUs send more messages. Hence, for the DSCA, the computation time for 40 ECUs and 200 messages is 849.78 ms and for 100 ECUs, the computation time is 4736.87 ms. For PSAC it is 513.47 ms and 1403.77 ms, while for our architecture, it is 408.77 ms and 1174.88 ms. Hence, the computation time of the proposed attribute-isolated architecture is less than that of the PSAC and DSCA.

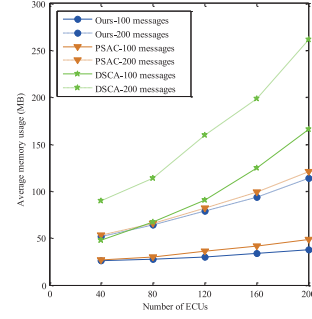


Fig. 12. Average memory storage.

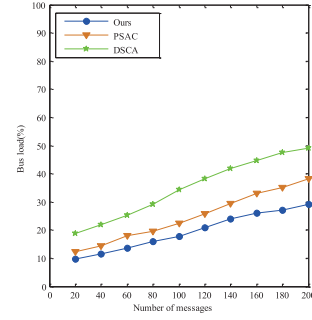


Fig. 13. Bus load comparisons.

2) *Average Storage Consumption*: The average memory usage behaves differently for the three architectures, as shown in Fig. 12. While for our architecture and PSAC, the average memory usage is nearly equal when the ECUs are added to the system, for the DSCA, the average memory usage increases linearly. This is because the DSCA needs to cache more authentication and encryption messages. For the DSCA, the average memory usage for 120 ECUs and 200 messages is 159.98 MB and for 200 ECUs, the average memory usage is 261.73 MB. For the PSAC, the average memory usage is 81.56 MB and 139.78 MB, while for our architecture, the average memory usage is 78.69 MB and 125.78 MB. Additionally, the DSCA requires more events than other architectures per new ECU, as a message exchange is more costly than our architecture or the PSAC and more ECUs mean more receivers. Hence the DSCA curve increases rapidly. For the PSAC, each ECU performs authentication and encryption and decryption to cache messages. The additional monitor information that results from more ECUs thus requires more receivers in the DSCA. For our attribute-isolated architecture, the curve slowly increases. The number of receivers is less than the number of ECUs in the vehicle and only specified ECUs can cache in-vehicle data. From this perspective, our architecture attains reasonable memory usage in comparison with the memory usage of the DSCA and PSAC.

3) *Bus Load*: The bus load rate is the sum of the bus percentages occupied by all data frames, and is an important indicator for measuring the communication performance of an in-vehicle network. We fixed the baud rate of the CAN-FD at 500 Kbps and evaluated the bus load for different cycles and number of messages. Fig. 13 shows the bus load of our architecture

compared to that of the PSAC and DSCA in terms of the number of messages. In the case of the DSCA, the bus load increase by approximately 50% due to the new message exchange and encryption. The bus load of the PSAC is slightly higher than that of our architecture since the PSAAC additionally has to transmit as many MAC messages as receivers. When there are 100 messages in the vehicle, the periods are 10 ms, 20 ms and 50 ms, the bus load of DSCA is 34.39%, that of PSAC is 22.34%, while the bus load of our architecture is 18.96%. Hence, the bus load of DSCA and PSAC are higher than that of our attribute isolated communication architecture. The proposed architecture can be well applied to the in-vehicle real-time environment.

VII. CONCLUSION

In this paper, a secure and efficient attribute-isolated automotive architecture was proposed. First, an analysis of the functional attributes of all of the in-vehicle ECUs in an intelligent connected environment and a division of the functional attributes of the ECUs into five classifications were performed. Second, based on the above-classified attributes, we demonstrated a secure attribute-isolated communication architecture. The ECUs have different access rights, allowing only the ECUs with the same functional attributes in the internal network of the vehicle to communicate. Then, it was proven that the proposed architecture could resist both forgery and eavesdropping attacks under the random oracle model. Finally, the secure attribute-isolated communication architecture was constructed in a hardware environment and evaluated with the IVNS. The evaluation results showed that the average memory usage with 120 ECUs and 100 messages is below 40 MB and the bus load can be reduced to 18.96% using the proposed security architecture compared with existing architectures. Our results confirm that the proposed architecture is suitable for an application to in-vehicle real-time environments.

In the future work, ICV will face driverless environments. These driverless cars can use nodes to perform edge computing and collect information for decision-making [30]. We will use gateways as nodes to enhance the collection capabilities of edge computing and ensure the efficiency and real-time performance of the automotive system. Therefore, securing automotive architecture based on edge computing will be the focus of our future research.

REFERENCES

- [1] W. Zeng, M. A. Khalid, and S. Chowdhury, "In-vehicle networks outlook: Achievements and challenges," *IEEE Commun. Surv. Tut.*, vol. 18, no. 3, pp. 1552–1571, Jan. 2017.
- [2] J.-S. Yang, H.-J. Lee, M.-W. Park, and J. Eom, "Security threats on national defense ICT based on iot," *Proc. Adv. Sci. Tech. Lett.*, vol. 97, pp. 94–98, Jun. 2015.
- [3] C. Miller and C. Valasek, "Remote exploitation of an unaltered passenger vehicle," *Proc. Black. Hat.*, vol. 2015, pp. 1–94, 2015.
- [4] S. Nie, L. Liu, and Y. Du, "Free-fall: HackingTesla from wireless to can bus," *Proc. Black. Hat.*, vol. 2017, pp. 1–16, 2017.
- [5] L. B. Othmane, H. Weffers, M. M. Mohamad, and M. Wolf, "A survey of security and privacy in connected vehicles," in *Proc. Wireless Sensor Mobile Ad-Hoc Netw.*, 2015, pp. 217–247.
- [6] K.-T. Cho and K. G. Shin, "Viden: Attacker identification on in-vehicle networks," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2017, pp. 1109–1123.
- [7] P. Subke, M. Moshref, A. Vach, and M. Steffebauer, "Measures to prevent unauthorized access to the in-vehicle e/e system, due to the security vulnerability of a remote diagnostic tester," *SAE Int. J. Cars. Elect. Syst.*, vol. 10, no. 2, pp. 422–429, Mar. 2017.
- [8] P. Mundhenk *et al.*, "Security in automotive networks: Lightweight authentication and authorization," *ACM Trans. Des. Automat. Electron. Syst.*, vol. 22, no. 2, pp. 1–27, Mar. 2017.
- [9] M. Wolf and A. Osterhues, "Safe messages modern cryptography protects automotive ecus," *ATZelektronik worldwide*, vol. 8, no. 2, pp. 38–43, Mar. 2013.
- [10] D. K. Nilsson, U. E. Larson, and E. Jonsson, "Efficient in-vehicle delayed data authentication based on compound message authentication codes," in *Proc. 68th IEEE Int. Conf. Veh. Technol.*, 2008, pp. 1–5.
- [11] B. Groza and P.-S. Murvay, "Broadcast authentication in a low speed controller area network," in *Proc. Int. Conf. E-Bus Telecommun.*, 2011, pp. 330–344.
- [12] B. Groza, S. Murvay, A. van Herrewwege, and I. Verbaauwhede, "Libra-can: A lightweight broadcast authentication protocol for controller area networks," in *Proc. Int. Conf. Cryptol. Netw. Secur.*, 2012, pp. 185–200.
- [13] B. Groza and S. Murvay, "Efficient protocols for secure broadcast in controller area networks," *IEEE Trans. Ind. Inform.*, vol. 9, no. 4, pp. 2034–2042, Nov. 2013.
- [14] J. Schmandt, A. T. Sherman, and N. Banerjee, "Mini-MAC: Raising the bar for vehicular security with a lightweight message authentication protocol," *Veh. Commun.*, vol. 9, pp. 188–196, Jul. 2017.
- [15] S. Tuohy, M. Glavin, C. Hughes, E. Jones, M. Trivedi, and L. Kilmartin, "Intra-vehicle networks: A review," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 534–545, Apr. 2014.
- [16] F. Hartwich *et al.*, "Can with flexible data-rate," in *Proc. Vector. Can., Inc.*, 2012, pp. 1–9.
- [17] 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 Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 993–1006, Apr. 2015.
- [18] S. Woo, H. J. Jo, I. S. Kim, and D. H. Lee, "A practical security architecture for in-vehicle CAN-FD," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 8, pp. 2248–2261, Aug. 2016.
- [19] C. Patsakis, K. Dellios, and M. Bouroche, "Towards a distributed secure in-vehicle communication architecture for modern vehicles," *Comput. Secur.*, vol. 40, pp. 60–74, Feb. 2014.
- [20] A. Rehman, M. M. Rathore, A. Paul, F. Saeed, and R. W. Ahmad, "Vehicular traffic optimisation and even distribution using ant colony in smart city environment," *IET Intell. Transp. Syst.*, vol. 12, no. 7, pp. 594–601, Sep. 2018.
- [21] D. Yang *et al.*, "Intelligent and connected vehicles: Current status and future perspectives," *Sci. China. Technol. Sci.*, vol. 61, no. 10, pp. 1446–1471, Sep. 2018.
- [22] B. Ran, H. Tan, J. Zhang, and Q. U. Xu, "Development status and trend of connected automated vehicle highway system," *J. Auto. Safe. Energy.*, vol. 9, no. 2, pp. 119–130, May. 2018.
- [23] E. Ohn-Bar and M. M. Trivedi, "Looking at humans in the age of self-driving and highly automated vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 1, no. 1, pp. 90–104, Jun. 2016.
- [24] A. Humayed, J. Lin, F. Li, and B. Luo, "Cyber-physical systems security-a survey," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1802–1831, May 2017.
- [25] L. Zhang and G. Orosz, "Motif-based design for connected vehicle systems in presence of heterogeneous connectivity structures and time delays," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 6, pp. 1638–1651, Jun. 2016.
- [26] J. Wang, D. Yang, and X. Lian, "Research on electrical/electronic architecture for connected vehicles," in *Proc. IET Int. Conf. Intell. Connected Veh.*, 2016, pp. 1–6.
- [27] M. Zhou, X. Qu, and S. Jin, "On the impact of cooperative autonomous vehicles in improving freeway merging: A modified intelligent driver model-based approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 6, pp. 1422–1428, Sep. 2017.
- [28] J. H. Kim, S.-H. Seo, N.-T. Hai, B. M. Cheon, Y. S. Lee, and J. W. Jeon, "Gateway framework for in-vehicle networks based on can, flexray, and ethernet," *IEEE Trans. Veh. Technol.*, vol. 64, no. 10, pp. 4472–4486, Oct. 2014.
- [29] P. Mundhenk, A. Mrowca, S. Steinhurst, M. Lukasiewicz, S. A. Fahmy, and S. Chakraborty, "Open source model and simulator for real-time performance analysis of automotive network security," *ACM Sig. Rev.*, vol. 13, no. 3, pp. 8–13, Jun. 2016.
- [30] W. Yu *et al.*, "A survey on the edge computing for the internet of things," *IEEE Access*, vol. 6, pp. 6900–6919, Nov. 2017.



Mu Han (Member, IEEE) was born in 1980. She received the Ph.D. degree from the School of Computer Science and Technology, Nanjing University of Science and Technology, China, in 2011. She is currently an Associate Professor with the School of Computer Science and Communication Engineering, Jiangsu University. Her research interests include cryptography, security and communication in vehicle network, the design of security protocols for smart car and Information Security, etc.



Fengwei Zhang received the Ph.D. degree in computer science from George Mason University. He is an Associate Professor with Department of Computer Science and Engineering, Southern University of Science and Technology. He was an Assistant Professor and the Director of the COMPASS Lab with Department of Computer Science, Wayne State University. His primary research interests include in the areas of systems security, with a focus on trustworthy execution, hardware-supported security, transparent malware analysis, and plausible deniability encryption.



Ailan Wan was born in Jiangsu Province, China. She received the B.S. degree from Jiangsu University, Zhenjiang, in 2016. She is currently working toward the M.S. degree with the Department of Computer Science and Communication Engineering, Jiangsu University. Her research interests include information security, cryptography, security of Electronic Control Unit in vehicular network, controller area network security.



Shidian Ma received the master's degree from the School of mechanical and automotive engineering, Hefei University of Technology, China, in 2005. He is currently an Associate Professor with School of Automotive Engineering Research Institute, Jiangsu University. His research interests include automotive electronic control technology, road traffic active safety prevention and control, security and communication of electronic control system.