Journal of Computers Vol. 32 No. 1, 2021, pp. 39-52
doi:10.3966/199115992021023201004

# An Information Security Protocol for Automotive Ethernet

Chu-Ting Wang, Gui-He Qin*, Rui Zhao, Shi-Min Song

College of Computer Science and Technology, Jilin University, Changchun 130012, China
wangct18@mails.jlu.edu.cn, qingh@jlu.edu.cn, zhaor1022@163.com, songsm18@mails.jlu.edu.cn

Received 4 November 2019; Revised 12 April 2020; Accepted 29 May 2020

Abstract. Automotive Ethernet is considered the backbone network of future vehicles owing to its high bandwidth, high throughput, and low cost. With the appearance of the connected car environment, in-vehicle networks (e.g., automotive Ethernet) are now connected to external networks (e.g., 3G/4G/5G mobile networks), enabling an attacker to perform an attack using automotive Ethernet vulnerabilities. Unfortunately, security problems have not been treated appropriately in automotive Ethernet. In this paper, we propose a security protocol for automotive Ethernet. The protocol has two secure modules: Key Distribution (KD) and Secure Communication (SC). During start-up phase, KD distributes keys to all legitimate ECUs. During the communication phase, SC provides the following important baseline security primitives: data confidentiality and data authenticity. We evaluate the effectiveness and real-time performance of the proposed security protocol using CANoe software and a MPC5646C microcontroller. Results show that the proposed security protocol can improve the defense ability of automotive Ethernet on the premise of meeting the real-time requirements.

Keywords: in-vehicle information security, automotive Ethernet, security protocol

## 1 Introduction

With the increase of vehicle functions, the amount of network data to be processed also grows. The communication capabilities of traditional in-vehicle networks such as Controller Area Network (CAN) and FlexRay are restricted, because of their limited bandwidth (maximum bandwidth of 1 Mbps for CAN and 10 Mbps for FlexRay) to handle the expected huge amount of sensor data. Meeting the real-time requirements in those networks especially for advanced driver assistance system applications is difficult [1].

Compared with traditional in-vehicle networks, automotive Ethernet has the advantages of high bandwidth, high throughput, and low cost. It is considered the next-generation in-vehicle network. Time-sensitive Networking and Time-triggered Ethernet are representative automotive Ethernet protocols that extend from traditional Ethernet. They are most suitable for automotive systems because they can guarantee the time. To date, many automotive manufacturers, such as BMW, have turned to automotive Ethernet as a complementary bus [2]. The application prospects of automotive Ethernet in vehicles are very broad.

With the development of connected car, vehicles need to open more and more interfaces to communicate with external networks. These interfaces greatly increase the risk of a vehicle being attacked. Attackers can attack vehicles through physical access (OBD-II, USB, charging piles), short-range wireless access (WIFI, Bluetooth, vehicle-borne radar) and long-range wireless access (radio, GPS, 3G/4G/5G) [3-6]. However, information security is not fully considered in the design of automotive Ethernet. Therefore, it is necessary to design a security mechanism while applying efficient automotive Ethernet.

Although there are many security mechanisms for traditional Ethernet, they have high computation and communication overhead. Automotive Ethernet is the communication medium of ECUs, which cannot finish a large number of computational tasks in a short time because of their limited computing

* Corresponding Author

power. These security mechanisms for traditional Ethernet cannot meet the real-time requirements of automotive Ethernet. Therefore, they are not suitable for direct application to automotive Ethernet. We must improve the defense capabilities of automotive Ethernet on the premise of meeting real-time requirements. A balance between security and real-time performance is necessary when designing security protocols for automotive Ethernet.

In this paper, we analyze the security requirements and constraints of automotive Ethernet. We propose an information security protocol for automotive Ethernet. The protocol has two secure modules: Key Distribution (KD) and Secure Communication (SC). KD achieves mutual authentication between the gateway ECU and all legitimate ECUs in the start-up phase. SC achieves the safety of data transmission in the communication phase. To evaluate the effectiveness and real-time performance of the proposed protocol, we construct an experimental platform based on a MPC5646C microcontroller. The main contributions of this paper are:

- Exploring the challenges in security in vehicles and analyzing the security requirements and design constraints of security protocols for automotive Ethernet
- Designing and developing a security protocol for automotive Ethernet under the premise of balancing effectiveness and real-time performance
- Analyzing the effectiveness and real-time performance of the proposed security protocol using CANoe software and a MPC5646C microcontroller

## 2 Related Works

The in-vehicle network used to be regarded as a closed network. With the continuous development of connected cars, the in-vehicle network is providing more useful services by being connected to external networks. However, such connection to external networks increases the vehicle attack surface, which makes in-vehicle networks vulnerable to cyberattacks, resulting in serious security problems [7]. Zhang et al. [8] showed that attackers can easily obtain and analyze data of vehicles and even control them by injecting messages through OBD-II. In terms of short-range wireless access, the most common way to attack vehicles is through Wi-Fi and Bluetooth. Josephlal et al. [9] demonstrated that attackers can easily extract the vulnerability of in-vehicle networks using various vulnerability scanning tools. Oka et al. [10] showed that attackers can force two communicating Bluetooth devices to repair and crack a PIN by listening for pairing information. In terms of long-range wireless access, attackers can easily attack vehicles through in-vehicle networks. Koscher et al. [11] stated that an attacker who is able to infiltrate virtually any ECU can exploit this ability to completely circumvent a broad series of safety-critical systems, for example, disabling brakes, braking individual wheels, and stopping an engine. Woo et al. [12] showed that attackers can inject forged data into in-vehicle networks through self-diagnostic application installed in a driver's mobile phone to control the critical actuator. Francillon et al. [13] found that attackers can control vehicles by means of intercepting messages transmitted between the vehicle system and the keys. With people's rising demand for vehicle functions, vehicles need to open an increasing number of interfaces to communicate with external networks, and consequently face more security threats.

To construct a secure in-vehicle network to deal with these threats, various studies and research projects have been carried out. In this trend, secure hardware architectures emerged. EVITA (E-Safety Vehicle Intrusion Protected Applications) analyzed use cases and defined security requirements for in-vehicle networks. Among these requirements, EVITA-MEDIUM-HSM was developed to provide a secure communication environment among ECUs [14]. In addition, as an important part of vehicle security hardware architecture, the security gateway was designed to guarantee the security of in-vehicle networks by encrypting messages between different systems [15]. However, they did not provide a specific security architecture for a particular communication protocol.

An important means to ensure the security of in-vehicle networks is to isolate it from external threats, and the automotive network firewall can exactly meet this requirement [16]. According to automotive requirements and adversary model, Pese et al. [17] discussed the partitioning of automotive network firewall's features in hardware and software and showed how to deploy an automotive embedded firewall system. Luo et al. [18] analyzed automotive security risks and designed security mechanisms on the basis of the network firewall. Their results showed that the automotive network firewall was effective and efficient. Although the automotive network firewall is able to prevent the access of attackers, it cannot

deal with internal threats, such as malicious hardware connected directly to the network or malicious apps installed in vehicles.

Compared with network firewall, intrusion detection is a network security technology that actively protects it-self from attacks. In recent years, automotive intrusion detection has attracted growing attention. The communication between ECUs of vehicles is usually orderly; hence, information entropy can be used to detect abnormal states. Song et al. [19] proposed an intrusion detection method for in-vehicle network on the basis of the analysis of time intervals of messages. The intrusion detection method was able to detect all message injection attacks. Wu et al. [20] proposed a sliding window anomaly detection method based on information entropy, which can provide real-time response to attacks. Although automotive intrusion has a good performance in the detection of intrusion behavior, it cannot block attacks and handle the effect of intrusion. Therefore, for in-vehicle networks, the best approach is to provide confidentiality and authenticity services directly.

Scholars used cryptographic methods to protect the information security of vehicles, such as confidentiality and authenticity. Groll et al. [21] analyzed the most important issues of in-vehicle network security risks, and proposed a flexible and adaptive solution based on trusted communication groups. The solution achieved confidential communication between different components of a vehicle. Herrewege et al. [22] proposed a message authentication protocol, which was simple and lightweight. The protocol can work on a CAN bus without any changes to existing nodes because it is backward compatible. Groza et al. [23] proposed a protocol for CAN entirely on the basis of simple symmetric primitives. It authenticates the sender on the receiving ECU with a mixed messages authentication code. Instead of authenticating separately for each node, they split the authentication keys between groups of nodes. The protocol was proven to be efficient especially if compromise nodes are in the minority. However, these scholars did not take full account of the security requirements of in-vehicle networks. Some of them only considered the confidentiality of data and did not authenticate the communication messages. And the others only considered the authenticity of the data, the confidentiality of the messages was ignored. Some scholars fully considered the above performance. For example, Hartkopp [24] considered network constraints, such as available resources and message length and proposed an authenticated protocol. The system consists of a time server and a key server. The time server broadcasts a timestamp at fixed period to guarantee the freshness of messages. The key server shares a symmetric long-term key with each legitimate ECU and coordinates the establishment of keys between communicating ECUs. The protocol has excellent safety protection capability, but it needs to introduce two new elements, the time server and the key server. It inevitably places additional burden on the system. Wang et al. [25] proposed the addition of a secure hardware-based module, or Security ECU onto the CAN bus, which can perform key distribution and message authentication, as well as destroy malicious messages before they are received by an ECU. Kurachi et al. [26] proposed an authentication method on the basis of a monitor node that authenticates other nodes in the network. The method overwrites data frames with error frames in real time to detect and destroy illegitimate data frames. However, according to this method, each vehicle needs to be equipped with an extra monitor node. If the monitor node is being attacked, then the whole network will also be affected. Although the above three methods can fully guarantee the confidentiality and authenticity of data, they need to introduce additional components. The introduction of additional components can lead to increased costs and a higher risk of a single point of failure in vehicles.

Without introducing additional components, Liu et al. [27] proposed a security protocol for on-board CAN, which was able to ensure the authenticity of the identity between the communicated nodes and authenticate messages. Nilsson et al. [28] proposed a data authentication method based on compound message authentication codes. The message authentication code is computed on the combination of successive messages and sent with subsequent messages. The receiver can verify the authenticity of the messages using the messages authentication code. This method can be used to provide effective security for communication data of in-vehicle networks. However, both of them only considered the limited data payload of CAN data frames, and did not take into account the processing of real-time data. Wang et al. [29] considered real-time performance and proposed an authentication mechanism that works on the basis of trust groups. A symmetric secret key is shared between high-trust groups. The mechanism implements authentication by sending a data message and then an authentication message. Although this authentication mechanism decreases the number of keys and has low delay, it cannot provide protection to the system if the compromised node is in high-trust groups. Hazem et al. [30] also considered real-time performance and proposed a lightweight broadcast authentication protocol that supported the processing

of real-time data. Instead of using message authentication codes, they use a 2-byte “magic number” that is calculated by applying the transformation function multiple times on an initial value. The receiver can verify the authenticity of the message by the “magic number”. The advantage of the protocol is that the overhead of authentication message exchange is minimal, but it uses only a 2-byte “magic number.” Therefore, it is not secure enough for actual use in automotive Ethernet.

In this paper, we balance the security and real-time performance of the automotive Ethernet, and propose a protocol that can be applied to the automotive Ethernet to provide security for vehicles from start-up to communication phase on the premise of meeting real-time requirements.

## 3 Security Requirements and Constraints

### 3.1 Security Requirements

Previous works have showed various types of possible attacks on vehicles. All these attacks ultimately stem from the vulnerabilities of in-vehicle networks [31-35]. Through the analysis of a large number of attack cases, we summarized two main vulnerabilities of automotive Ethernet: the lack of encryption and the lack of authentication. To construct a secure automotive Ethernet, these vulnerabilities need to be eliminated. Therefore, encrypting and authenticating data frames are necessary to prevent forgery attack and replay attack. We identify the requirements to provide a secure automotive Ethernet as follows:

- *Confidentiality*: In many applications, a large number of safety-critical messages are transmitted among ECUs [36]. An attacker who is able to eavesdrop data frames can easily obtain important information. Therefore, encrypting every data frame in automotive Ethernet is necessary to provide confidentiality. The plaintext form of the data frame should be attainable only to a legitimate ECU.
- *Authenticity*: Message authentication is crucial for many applications in automotive Ethernet. An attacker can easily inject messages; hence, the receiver needs to ensure that the data used in any decision-making process is from the legitimate sender. Message authentication allows the receiver to confirm that the message was sent by the legitimated sender. Moreover, all data transmitted in the automotive Ethernet are time-varying data. Thus, we must ensure that each message is fresh, that data are recent, and that no attacker has replayed old messages.

### 3.2 Design Constraints

When designing a security protocol for automotive Ethernet, we usually need to consider two constraints:

- *Resource limited node*: Owing to cost considerations, the processing and storage capabilities of the ECU are limited. For example, the S12XD series of microcontrollers, which are produced by Freescale, provide up to 512 KB of flash memory, 32 KB of RAM and 40 KB of EEPROM with a core operating frequency of 80 Hz [37]. In addition to software updates, flash memory is usually not written. Thus, EEPROM holds a large number of nonvolatile application data. Buffering and storage consume space in RAM, which is rarer and more expensive than flash memory. Therefore, a method that requires a large number of processing or storage capabilities is not feasible for an in-vehicle system.
- *Real-time deadlines*: In an in-vehicle system, most processes must be completed within specific deadlines (typically in milliseconds). A specific method must be completed within a known time bound, that is fast enough to match the physical time constants of the system being controlled. If the completion time of the control application exceeds the deadline, then the control result may not have the expected effect or even cause serious consequences. Therefore, the ECU must complete the authentication of messages as quickly as possible to meet the real-time requirements of the control applications.

To achieve these security requirements while satisfying the constraints, we designed and implemented two security modules: KD and SC. KD distributes keys to all legitimate ECUs during the start-up phase, whereas SC provides data confidentiality and data authenticity. Table 1 provides a list of the notation used in this paper.

**Table 1.** Notation used for the proposed protocol

| Notation | Description |
|---|---|
| $ECU_i$ | ECU using identity i |
| $ECU_j$ | ECU using identity j |
| $ECU_G$ | Gateway ECU |
| $ID_i$ | Identity of $ECU_i$ |
| $PUK_G$ | Public key of $ECU_G$ |
| $PRK_G$ | Private key of $ECU_G$ |
| $PUK_i$ | Public key of $ECU_i$ |
| $PRK_i$ | Private key of $ECU_i$ |
| $C$ | Ciphertext |
| $M$ | Plaintext |
| $RN_i$ | Random number selected by $ECU_G$ when distributing keys for $ECU_i$ |
| $K_1$ | Encryption key |
| $K_2$ | Authentication key |
| $SN_i$ | Serial number maintained by $ECU_i$ |
| $SN_j$ | Serial number maintained by $ECU_j$ |
| $MAC_G$ | Message authentication code generated by $ECU_G$ |
| $MAC_i$ | Message authentication code generated by $ECU_i$ |
| $MAC_j$ | Message authentication code generated by $ECU_j$ |
| $SE_x$ | Symmetric encryption and decryption function using x |
| $AE_x$ | Asymmetric encryption function using x |
| $AD_x$ | Asymmetric decryption function using x |
| $H_x$ | Keyed hash function using x |

# 4 Key Distribution Module

## 4.1 Implementation Process of KD

After a vehicle is started, the gateway ECU distributes keys to each legitimate ECU in a fixed order. The gate-way ECU has a set of digital certificates that contain information, such as identity and keys of each legitimate ECU in the network. On the basis of asymmetric encryption algorithm RSA and dynamic cipher mechanism, we construct a secure and efficient key distribution process in the automotive Ethernet environment, as this process provides mutual identity authentication and implicit key distribution.

The key distribution process is shown in Fig. 1.

*Stage 1*: $ECU_G$ selects a random number $RN_i$, encrypts the combination of $K_1$, $K_2$, and $RN_i$ using $PUK_i$ to obtain $C_1$ and transmits it to $ECU_i$

$$C_1 = AE_{PUK_i}\left(K_1 \| K_2 \| RN_i\right). \tag{1}$$

*Stage 2*: $ECU_i$ decrypts $C_1$ using $PRK_i$ to obtain $K_1$, $K_2$, and $RN_i$

$$K_1 \| K_2 \| RN_i = AD_{PRK_i}\left(C_1\right). \tag{2}$$

*Stage 3*: $ECU_i$ generates $MAC_i$ for $ID_i$ and $RN_i$ with $K_2$ and transmits it to $ECU_G$ with $ID_i$

$$MAC_i = H_{K_2}\left(ID_i \| RN_i\right). \tag{3}$$

*Stage 4*: $ECU_G$ generates $MAC_G$ for $ID_i$ and $RN_i$ with $K_2$

$$MAC_G = H_{K_2}\left(ID_i \| RN_i\right). \tag{4}$$

*Stage 5*: $ECU_G$ compares $MAC_G$ with $MAC_i$. By comparing them, we can determine whether the key distribution for $ECU_i$ is successful.

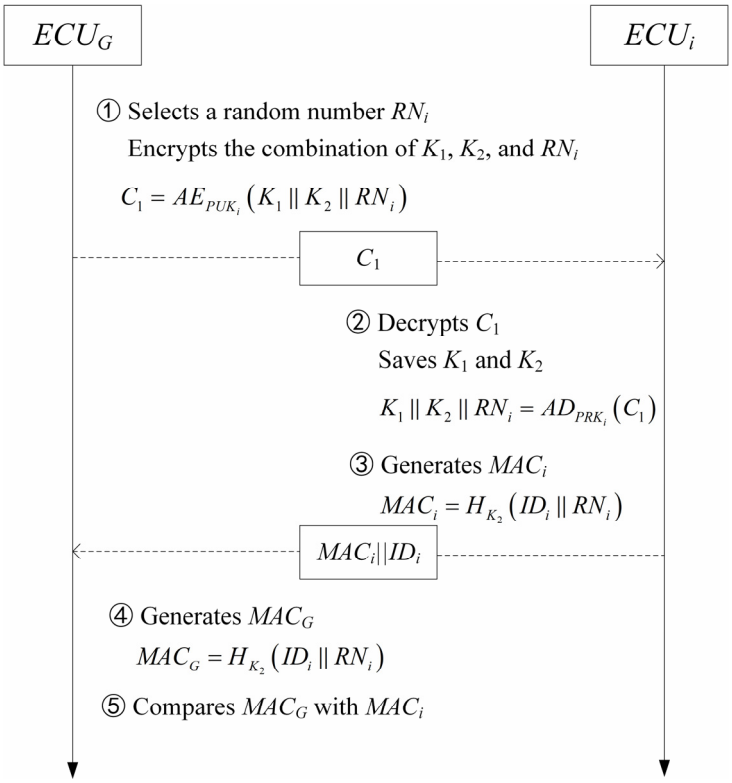


**Fig. 1.** Key distribution process

After the key distribution process is completed, all legitimate ECUs successfully obtain the encryption key and the authentication key. The keys are valid only after this start-up and periodically updated by the gateway ECU.

### 4.2 Security Analysis of KD

In the start-up phase, the gateway ECU performs a key distribution process for each ECU. Although an attacker can eavesdrop the key distribution message, it does not have the private key of the legitimate ECU. Therefore, it cannot decrypt the message containing the keys. The attacker cannot obtain the keys that are required for the communication process. Moreover, the public key is shared only between the legitimate ECU and the gateway ECU. The attacker who does not have the public key cannot open the key distribution process by disguising as the gateway ECU.

Given that the security of RSA has been proven in [38], an attacker cannot affect the key distribution process without the public and private keys.

## 5 Secure Communication Module

### 5.1 Implementation Process of SC

After the key distribution process is completed, each ECU obtains the encryption key and the authentication key. Hence, they have the ability to encrypt, decrypt, and authenticate messages.

We comprehensively analyze the security requirements and constraints of automotive Ethernet. We choose symmetric encryption and decryption algorithm DES and message digest algorithm HMAC-MD5 as the main algorithm to design a secure communication method. In this method, each ECU has a serial number, which is used to provide the freshness of the message. Before communicating, the initial serial number is set to zero, and is incremented each time it is used. Each message in the network has a string of message authentication code calculated by the serial number and the encrypted data. The receiver verifies the authenticity of messages by comparing message authentication codes.

The communication process between the sender $ECU_i$ and the receiver $ECU_j$ is shown in Fig. 2.

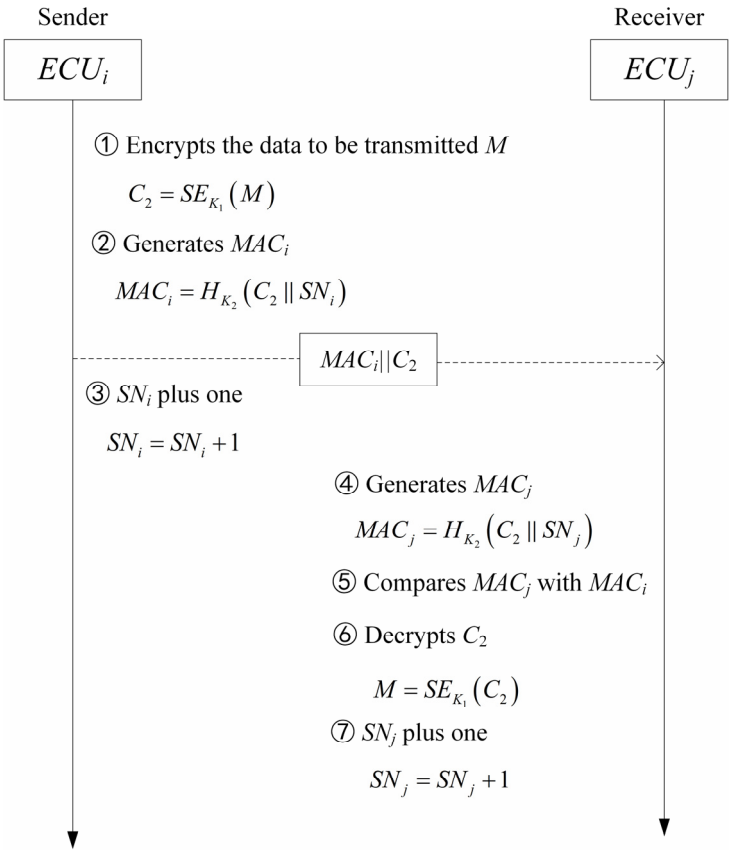


**Fig. 2.** Secure communication process

Send messages:
*Stage 1*: Sender $ECU_i$ encrypts the data to be transmitted $M$ with $K_1$ to obtain $C_2$

$$C_2 = SE_{K_1}(M). \tag{5}$$

*Stage 2*: $ECU_i$ generates $MAC_i$ for $C_2$ and $SN_i$ with $K_2$ and transmits it to $ECU_j$ with $C_2$

$$MAC_i = H_{K_2}(C_2 \| SN_i). \tag{6}$$

*Stage 3*: $ECU_i$ makes $SN_i$ plus one

$$SN_i = SN_i + 1. \tag{7}$$

Receive messages:
*Stage 1*: Receiver $ECU_j$ generates $MAC_j$ for $C_2$ and $SN_j$ with $K_2$

$$MAC_j = H_{K_2}(C_2 \| SN_j). \tag{8}$$

*Stage 2*: $ECU_i$ compares $MAC_j$ with $MAC_i$. By comparing them, we can confirm whether the message is legitimate. If the message is legitimate, then the execution continues. Otherwise, an attack alarm is set off

*Stage 3*: $ECU_j$ decrypts $C_2$ using $K_1$ to obtain $M$

$$M = SE_{K_1}(C_2). \tag{9}$$

*Stage 4*: $ECU_j$ makes $SN_j$ plus one

$$SN_j = SN_j + 1. \tag{10}$$

### 5.2 Security Analysis of SC

In the communication phase, the sender encrypts the data to be transmitted with the encryption key $K_1$, and the receiver decrypts the received data with the same encryption key $K_1$. Although an attacker can eavesdrop the communication messages between the legitimate ECUs, the attacker does not have the encryption key $K_1$. Therefore, it cannot decrypt the communication messages, that is, it cannot obtain the plaintext form of the messages.

During the communication, the 128-bit MAC generated by the HMAC-MD5 provides the authenticity of messages, and the serial number maintained by the sender and the receiver provides the freshness of the messages.

The sender generates the MAC for the message to be transmitted and the serial number maintained by the sender with the authentication key $K_2$. The receiver also generates the MAC for the message received and the serial number maintained by the receiver with the authentication key $K_2$. By comparing the MACs, the receiver can confirm whether the message is legitimate. An attacker can also possibly use the known structure of the input to a MAC to generate forged messages. However, an attacker cannot generate MAC corresponding to the forged message without the authentication key. The only way for an attacker to forge a MAC is to select a 128-bit string from $2^{128}$ possible MACs. In the in-vehicle system, the time interval of the message transmission is short. Hence, any attacker in the network can forge a 128-bit MAC in a short time, but if it transmits one data frame every 5 ms, the time it takes to transmit $2^{128}$ data frames is incalculable. Therefore, the chances that an attacker attacks successfully by forging the MAC are close to zero.

Given that the security of DES and HMAC-MD5 have been proven in [39] and [40], the attacker cannot crack the ciphertext of messages or participate in the communication process by disguising as a legitimate ECU.

## 6 Evaluation

For effectiveness and real-time performance evaluation of the proposed security protocol, we construct an automotive Ethernet experimental platform. As shown in Fig. 3, the experimental platform includes a gateway ECU and nine common ECUs, among which $ECU_9$ is illegitimate and the others are legitimate.

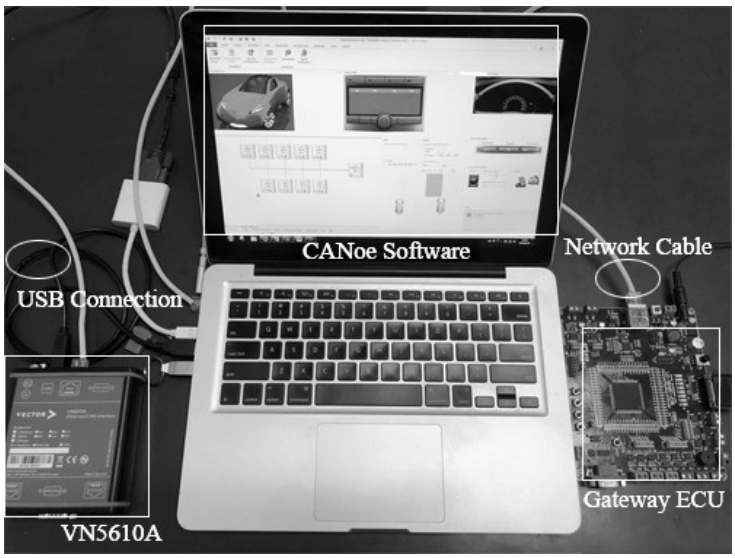


**Fig. 3.** Experimental platform

### 6.1 Effectiveness Evaluation of KD

Table 2 shows the process of key distribution. The illegitimate $ECU_9$ attacks the network in the following two cases:

- The $ECU_9$ attempts to disguise as a legitimate ECU to obtain the keys. Given that the key assignment message sent by the $ECU_G$ is encrypted by the public key of the legitimate ECU, the $ECU_9$ does not have a private key corresponding to the public key. Therefore, even if it intercepts the message, it cannot obtain the keys.
- The $ECU_9$ attempts to intercept a confirmation message returned by a legitimate ECU to the $ECU_G$, and transmits it again to cause confusion in the key distribution process, as the 6th message in Table 2 shows. The message authentication code in the confirmation message is generated for the identification of the legitimate ECU and the random number selected by the $ECU_G$. At this time, the random number has been updated. Therefore, the $ECU_G$ can confirm whether the message is from a legitimate ECU by verifying the message authentication code.

**Table 2.** Experimental data of Key Distribution

| | Source address | Destination address | Length of information | Information (first 8 bytes) |
|---|---|---|---|---|
| 1 | 00-04-9f-00-00-00 | 00-04-9f-00-00-01 | 0x0040 | 45 78 4D 36 69 59 71 55 |
| 2 | 00-04-9f-00-00-01 | 00-04-9f-00-00-00 | 0x0040 | 01 45 79 75 53 39 43 30 |
| 3 | 00-04-9f-00-00-00 | 00-04-9f-00-00-02 | 0x0040 | 65 68 57 42 51 45 67 44 |
| 4 | 00-04-9f-00-00-02 | 00-04-9f-00-00-00 | 0x0040 | 02 38 77 76 42 66 75 43 |
| 5 | 00-04-9f-00-00-00 | 00-04-9f-00-00-03 | 0x0040 | 6a 68 72 44 70 4f 6e 4a |
| 6 | 00-04-9f-00-00-09 | 00-04-9f-00-00-00 | 0x0040 | 02 38 77 76 42 66 75 43 |
| 7 | 00-04-9f-00-00-03 | 00-04-9f-00-00-00 | 0x0040 | 03 64 61 6a 6c 78 61 57 |
| 8 | 00-04-9f-00-00-00 | 00-04-9f-00-00-04 | 0x0040 | 51 51 58 72 4b 33 6d 38 |
| 9 | 00-04-9f-00-00-04 | 00-04-9f-00-00-00 | 0x0040 | 04 30 50 59 62 47 6d 58 |
| 10 | 00-04-9f-00-00-00 | 00-04-9f-00-00-05 | 0x0040 | 34 72 45 6c 79 64 71 2b |
| 11 | 00-04-9f-00-00-05 | 00-04-9f-00-00-00 | 0x0040 | 05 30 34 73 66 57 30 58 |
| 12 | 00-04-9f-00-00-00 | 00-04-9f-00-00-06 | 0x0040 | 41 51 66 51 36 31 6c 47 |
| 13 | 00-04-9f-00-00-06 | 00-04-9f-00-00-00 | 0x0040 | 06 77 6f 71 4a 61 50 73 |
| 14 | 00-04-9f-00-00-00 | 00-04-9f-00-00-07 | 0x0040 | 55 32 46 73 64 47 56 6b |
| 15 | 00-04-9f-00-00-07 | 00-04-9f-00-00-00 | 0x0040 | 07 75 65 70 6b 33 30 77 |
| 16 | 00-04-9f-00-00-00 | 00-04-9f-00-00-08 | 0x0040 | 54 67 6d 38 6e 4e 75 71 |
| 17 | 00-04-9f-00-00-08 | 00-04-9f-00-00-00 | 0x0040 | 08 68 57 42 51 45 67 44 |

### 6.2 Effectiveness Evaluation of SC

Table 3 shows the process of communication between all ECUs. The illegitimate $ECU_9$ attacks the $ECU_1$ in the following three cases:

- *Eavesdrop messages:* The $ECU_9$ attempts to obtain important information by eavesdropping messages in the network. However, each legitimate ECU in the network has an encryption key and the message is encrypted using the encryption key before transmission. Hence, the $ECU_9$ cannot decrypt the message even if it receives the message.
- *Forge messages:* The $ECU_9$ attempts to communicate with the $ECU_1$ by disguising as a legitimate ECU, but it cannot generate the correct message authentication code without the authentication key. As the 17th to the 18th messages in Table 3 show, the $ECU_9$ transmits a message with the incorrect message authentication code. After receiving the message, the $ECU_1$ verifies the message authentication code and sets off an attack alert.
- *Replay messages:* The $ECU_9$ attempts to make the network unable to work normally by transmitting the message that has been previously transmitted. In the network, both sides of the communication record the number of messages by maintaining the serial number. As the 19th to the 20th messages in Table 3 show, the $ECU_9$ transmits a message that has been transmitted. At this time, the serial number of the $ECU_1$ has changed; hence, the correct message authentication code has also changed. After receiving the message, the $ECU_1$ verifies the message authentication code and sets off an attack alert.

**Table 3.** Experimental data of Secure Communication

| | Source address | Destination address | Length of information | Information (first 8 bytes) | Legal message |
|---|---|---|---|---|---|
| 1 | 00-04-9f-00-00-01 | 00-04-9f-00-00-02 | 0x05dc | 41 42 48 30 64 74 61 48 | True |
| 2 | 00-04-9f-00-00-01 | 00-04-9f-00-00-02 | 0x05dc | 4c 56 4e 4d 6c 32 6c 33 | True |
| 3 | 00-04-9f-00-00-02 | 00-04-9f-00-00-01 | 0x05dc | 36 67 6c 36 78 67 49 72 | True |
| 4 | 00-04-9f-00-00-02 | 00-04-9f-00-00-01 | 0x05dc | 50 6c 59 6b 58 50 34 34 | True |
| 5 | 00-04-9f-00-00-03 | 00-04-9f-00-00-04 | 0x05dc | 55 32 46 73 64 47 56 6b | True |
| 6 | 00-04-9f-00-00-03 | 00-04-9f-00-00-04 | 0x05dc | 58 31 39 4d 6d 74 4d 79 | True |
| 7 | 00-04-9f-00-00-04 | 00-04-9f-00-00-03 | 0x05dc | 46 45 36 4c 72 69 58 4c | True |
| 8 | 00-04-9f-00-00-04 | 00-04-9f-00-00-03 | 0x05dc | 65 54 4b 75 7a 75 68 66 | True |
| 9 | 00-04-9f-00-00-05 | 00-04-9f-00-00-06 | 0x05dc | 4f 53 69 32 6d 46 45 30 | True |
| 10 | 00-04-9f-00-00-05 | 00-04-9f-00-00-06 | 0x05dc | 54 48 59 75 24 93 56 33 | True |
| 11 | 00-04-9f-00-00-06 | 00-04-9f-00-00-05 | 0x05dc | 48 59 79 6f 4d 52 32 73 | True |
| 12 | 00-04-9f-00-00-06 | 00-04-9f-00-00-05 | 0x05dc | 44 67 51 71 4c 4a 48 37 | True |
| 13 | 00-04-9f-00-00-07 | 00-04-9f-00-00-08 | 0x05dc | 5a 38 77 76 42 66 75 43 | True |
| 14 | 00-04-9f-00-00-07 | 00-04-9f-00-00-08 | 0x05dc | 32 71 4c 33 6f 4e 79 4d | True |
| 15 | 00-04-9f-00-00-08 | 00-04-9f-00-00-07 | 0x05dc | 64 81 26 45 78 63 21 49 | True |
| 16 | 00-04-9f-00-00-08 | 00-04-9f-00-00-07 | 0x05dc | 2a 25 8e 64 59 22 2f 9d | True |
| 17 | 00-04-9f-00-00-09 | 00-04-9f-00-00-01 | 0x05dc | 54 46 55 1a 23 25 6b 3f | False |
| 18 | 00-04-9f-00-00-09 | 00-04-9f-00-00-01 | 0x05dc | b4 b4 43 76 c4 f5 65 23 | False |
| 19 | 00-04-9f-00-00-09 | 00-04-9f-00-00-01 | 0x05dc | 4c 56 4e 4d 6c 32 6c 33 | False |
| 20 | 00-04-9f-00-00-09 | 00-04-9f-00-00-01 | 0x05dc | 50 6c 59 6b 58 50 34 34 | False |

### 6.3 Real-time Performance Evaluation

The KD adopts the asymmetric encryption algorithm RSA and the message digest algorithm HMAC-MD5. It includes five parts of time cost, which are the time cost of encrypting messages by the sender, transmitting data frames in the network, decrypting messages by the receiver, generating the message authentication codes by the sender, and generating the message authentication codes by the receiver. Therefore, the time cost $T_a$ of the key distribution for a single legitimate ECU in the KD is:

$$T_a = 2 * \left( T_{RSA} + T_{HMAC-MD5} + T_{com} \right). \tag{11}$$

$T_{RSA}$ is the time cost generated by the asymmetric encryption algorithm RSA in encrypting or decrypting a message, $T_{HMAC\text{-}MD5}$ is the time cost generated by the message digest algorithm HMAC-MD5 in calculating a message authentication code, and $T_{com}$ is the time cost generated by the transmission of a data frame.

The SC adopts the symmetric encryption algorithm DES and the message digest algorithm HMAC-MD5. It includes five parts of time cost, which are the time cost of encrypting messages by the sender, generating the message authentication codes by the sender, transmitting data frames in the network, decrypting the messages by the receiver, and generating the message authentication codes by the receiver. Therefore, the communication response time $T_r$ in the SC is:

$$T_r = \left( T_{DES} + T_{HMAC-MD5} \right) * 2 + T_{com}. \tag{12}$$

$T_{DES}$ is the time cost generated by the symmetric encryption algorithm DES in encrypting or decrypting a message, $T_{HMAC\text{-}MD5}$ is the time cost generated by the message digest algorithm HMAC-MD5 in calculating a message authentication code, and $T_{com}$ is the time cost generated by the transmission of a data frame.

In the experiment, we measured the execution times of RSA algorithm, DES algorithm, and HMAC-MD5 algorithm by implementing the algorithms on MPC5646C. The execution time of the algorithm is closely related to the CPU clock rate. Therefore, the CPU clock rate was changed to 60, 80, 100, and 120 MHz. For reliable results, we repeated the measurement 10,000 times to obtain the average execution time. The results are shown in Fig. 4. We also implemented these algorithms on TriCore and FPGA. The CPU clock rate was changed to 80 MHz. The results are shown in Fig. 5. If the proposed protocol is implemented on another platform, then it will perform even better.

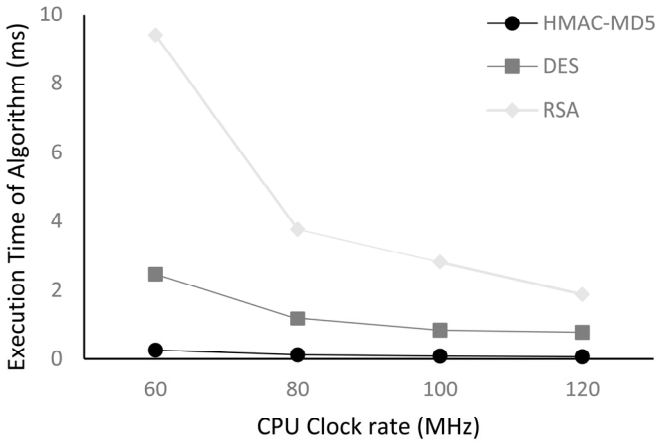


**Fig. 4.** Execution time of algorithm

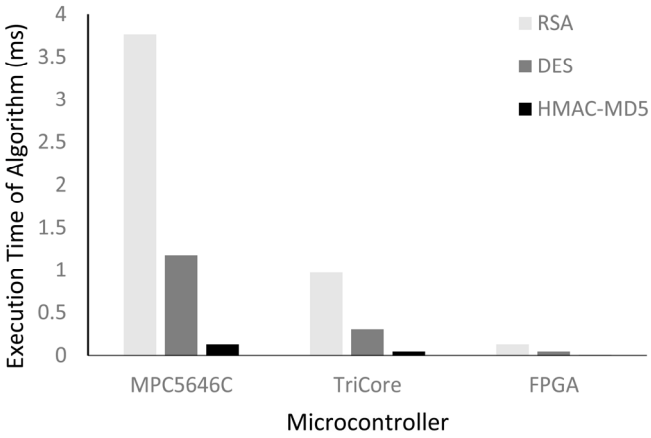


**Fig. 5.** Execution time of algorithm on different microcontroller

We tested the effect of the proposed protocol on in-vehicle systems of different scales. As the scale of the vehicle system grows, the number of ECU also increases. To evaluate the performance of the proposed protocol on different system scales, for each CPU clock rate, we plotted the key distribution time and the average communication response time in terms of the number of ECUs. As shown in Fig. 6 and Fig. 7, the horizontal axis represents the number of ECUs, which is varied in the range of 10, 20, 30 and 40. That is, the number of ECUs on the horizontal axis represents the scale of the in-vehicle system.

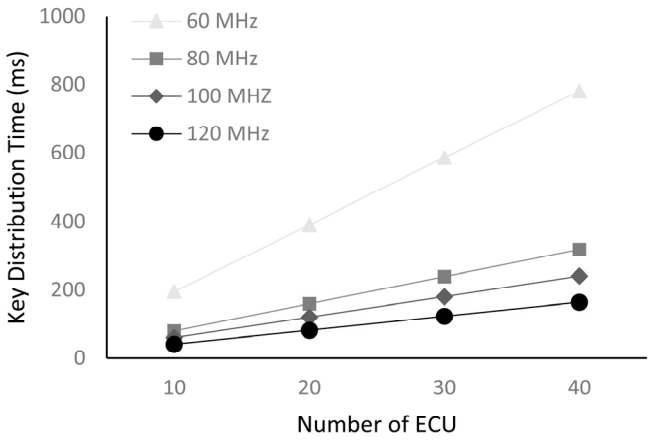


**Fig. 6.** Key distribution time

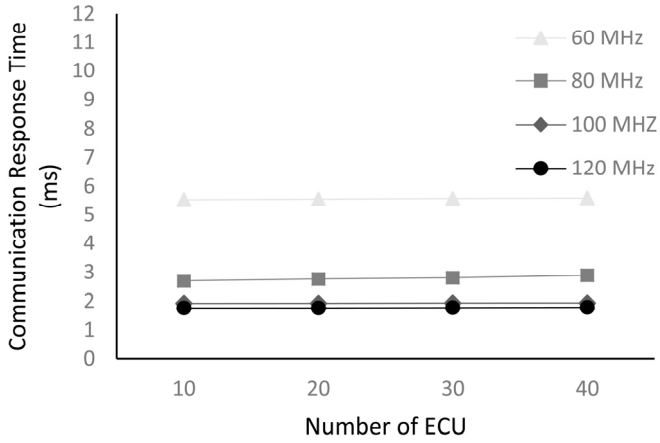


**Fig. 7.** Communication response time

During start-up, the gateway ECU distributes keys to each legitimate ECU in a fixed order. Therefore, the number of ECUs has a great influence on the key distribution time. Fig. 6 shows that at any CPU clock rate, as the number of ECUs increases, so does the key distribution time. When the CPU clock rate is 120 MHz, 10 ECUs completed the key distribution process in less than 40 ms. When the CPU clock rate is 60 MHz, 40 ECUs completed the key distribution process in less than 800 ms. For a real automotive Ethernet, the CPU clock rate represents vehicle performance. Therefore, when applying the proposed security protocol to vehicles with high-performance or low-performance ECUs, large or small system scale, it is able to complete the key distribution process in an acceptable time. In addition, the communication response time is also important. As shown in Fig. 7, when the CPU clock rate is 120 MHz, the communication response time is less than 2 ms. When the CPU clock rate is 60 MHz, the communication response time is less than 6 ms. Moreover, as the number of ECU increases, the communication response time remains stable without much change. Even if there is a change, it is so small that will not have any effect on the in-vehicle system. Therefore, when applying the proposed security protocol to vehicles with high-performance or low-performance ECUs, large or small system scale, the communication response time can meet the real-time requirements of automotive Ethernet. Therefore, when applying the proposed security protocol to vehicles, its availability can be fully guaranteed.

## 7 Conclusions

In this paper, we summarized two main vulnerabilities of automotive Ethernet and identified the requirements to construct a secure automotive Ethernet. According to the requirements, we proposed a security protocol that could be applied to the automotive Ethernet to provide security for vehicles from start-up to the communication phase. We theoretically analyze the security of the proposed protocol. Furthermore, we evaluated the effectiveness and real-time performance of the proposed security protocol through an evaluation based on MPC5646C microcontroller and CANoe. Regardless of vehicle performance or system scale, the proposed protocol could perform well. In the proposed protocol, the key distribution relies on asymmetric mechanism. Compared with asymmetric mechanisms, symmetric mechanisms have low computation, communication, and storage overhead. In the future, we plan to improve the performance of the proposed protocol with an implementation of the symmetric mechanism instead of asymmetric mechanism in the key distribution.

## References

[1] A. Diarra, A. Zimmermann, System design issues for future in-vehicle Ethernet-based time and safety critical networks, in: Proc. IEEE International Systems Conference, 2015.

[2] G. Malaguti, M. Dian, C. Ferraresi, M. Ruggeri, Comparison on technological opportunities for in-vehicle Ethernet networks, in: Proc. International Conference on Industrial Informatics, 2013.

[3] J. Lastinec, M. Keszeli, Analysis of realistic attack scenarios in vehicle ad-hoc networks, in: Proc. International Symposium on Digital Forensics and Security, 2019.

[4] P. Kleberger, T. Olovsson, E. Jonsson, Security aspects of the in-vehicle network in the connected car, in: Proc. IEEE Intelligent Vehicles Symposium, 2011.

[5] T. Hoppe, S. Kiltz, J. Dittmann, Security threats to automotive CAN networks—Practical examples and selected short-term countermeasures, in Proc. International Conference on Computer Safety, 2008.

[6] K. Mawonde, B. Isong, F. Lugayizi, A. Abu-Mahfouz, A survey on vehicle security systems: Approaches and technologies, in: Proc. Conference of the IEEE Industrial Electronics Society, 2018.

[7] Y. Zhang, S. Han, S. Zhong, P. Shi, X. Shao, Research on Information Security Test Evaluation Method Based on Intelligent Connected Vehicle, in: Proc. Security and Privacy in New Computing Environments - 2nd EAI International Conference, 2019.

[8] Y. Zhang, B. Ge, X. Li, B. Shi, B. Li, Controlling a Car Through OBD Injection, in: Proc. 3rd IEEE International Conference on Cyber Security and Cloud Computing, 2016.

[9] E. F. M. Josephlal, S. Adepu, Vulnerability analysis of an automotive infotainment system's WIFI capability, in: Proc. 19th IEEE International Symposium on High Assurance Systems Engineering, 2019.

[10] D. K. Oka, T. Furue, L. Langenhop, T. Nishimura, Survey of vehicle IoT bluetooth devices, in: Proc. IEEE 7th International Conference on Service-Oriented Computing and Applications, 2014.

[11] K. Koscher, A. Czekis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Snacham, S. Savage, Experimental security analysis of a modern automobile, in: Proc. IEEE Symposium on Security and Privacy, 2010.

[12] S. Woo, H. Jo, D. Lee, A Practical Wireless Attack on the Connected Car and Security Protocol for In-Vehicle CAN, IEEE Transactions on Intelligent Transportation Systems 16(2)(2015) 993-1006.

[13] A. Francillon, B. Danev, S. Capkun, Relay Attacks on Passive Keyless Entry and Start Systems in Modern Car, in: Proc. the Network and Distributed System Security Symposium, 2011.

[14] H. Olaf, E-safety vehicle intrusion protected application. <http://www.evita-project.org/Publications/EVITAD0.pdf>, 2008 (accessed 19.09.01)

[15] F. Luo, Q. Hu, Security Mechanisms Design for In-Vehicle Network Gateway, in: Proc. SAE World Congress Experience, 2018.

[16] K. Schmidit, H. Zweck, U. Dannebaum, Hardware and Software Constraints for Automotive Firewall System, in: Proc. SAE World Congress and Exhibition, 2016.

[17] M. D. Pese, K. Schmidt, H. Zweck, Hardware/Software Co-Design of an Automotive Embedded Firewall, in: Proc. SAE World Congress Experience, 2017.

[18] F. Luo, S. Hou, Security mechanisms design of automotive gateway firewall, in: Proc. SAE World Congress Experience, 2019.

[19] H. M. Song, H. R. Kim, H. K. Kim, Intrusion detection system based on the analysis of time intervals of CAN messages for in-vehicle network, in: Proc. 30th International Conference on Information Networking, 2016.

[20] W. Wu, Y. Huang, R. Kurachi, G. Zeng, G. Xie, R. Li, K. Li, Sliding Window Optimized Information Entropy Analysis Method for Intrusion Detection on In-Vehicle Networks 6(2018) 45233-45245.

[21] A. Groll, C. Ruland, Secure and authentic communication on existing in-vehicle networks, in: Proc. 2009 IEEE Intelligent Vehicles Symposium, 2009.

[22] A. V. Herrewege, D. Singelee, I. Verbauwhede, CANAuth - A Simple, Backward Compatible Broadcast Authentication Protocol for CAN bus, in: Proc. 9th International conference on Embedded Security in Cars, 2011.

[23] B. Groza, S. Murvay, H. Van, Anthony, I. Verbauwhede, LiBrA-CAN: A lightweight broadcast authentication protocol for controller area networks, in: Proc. 11th International Conference on Cryptology and Network Security, 2012.

[24] O. Hartkopp, C. Reuber, R. Schilling, MaCAN - Message Authenticated CAN, in: Proc. 10th International conference on Embedded Security in Cars, 2012.

[25] E, Wang, W. Xu, S. Sastry, S. Liu, K. Zeng, Hardware module-based message authentication in intra-vehicle networks, in: Proc. International Conference on Cyber-Physical Systems, 2017.

[26] R. Kurachi, Y. Matsubara, H. Takada, N. Adachi, Y. Miyashita, S. Horihata, CaCAN - Centralized Authentication System in CAN, in: Proc. 12th International conference on Embedded Security in Cars, 2014.

[27] Y. Liu, G. Qin, R. Zhao, Security Protocol for On-Board Controller Area Network, Journal of Xi'an Jiaotong University 52(5)(2018) 94-100.

[28] D.K. Nilsson, U. Larson, E. Jonsson, Efficient In-Vehicle Delayed Data Authentication Based on Compound Message Authentication Code, in: Proc. 68th Semi-Annual IEEE Vehicular Technology, 2008.

[29] Q. Wang, S. Sawhney, VeCure: A practical security framework to protect the CAN bus of vehicles, in: Proc. International Conference on the Internet of Things, 2014.

[30] A. Hazem, H. A. Fahmy, LCAP - A Lightweight CAN Authentication Protocol for securing in-vehicle networks, in: Proc. 10th International conference on Embedded Security in Cars, 2012.

[31] J. Choi, S. Jin, Security Threats in Connected Car Environment and Proposal of In-Vehicle Infotainment-Based Access Control Mechanism, in: Proc. International Conference on Future Information Technology, 2018.

[32] L, Baldanzi, L. Crocetti, M. Bertolucci, L. Fanucci, S. Saponara, Analysis of cybersecurity weakness in automotive in-vehicle networking and hardware accelerators for real-time cryptography, in: Proc. International Conference on Applications in Electronics Pervading Industry, 2018.

[33] W, Xiong, F. Krantz, R. Lagerstrom, Threat modeling and attack simulations of connected vehicles: A research outlook, in: Proc. International Conference on Information Systems Security and Privacy, 2018.

[34] K. Zaidi, M. Rajarajanm, Vehicular internet: Security & privacy challenges and opportunities. Future Internet 7(3)(2015) 275-275.

[35] Y. Sun, L. Wu, S. Wu, S. Li, T. Zhang, J. Xu, Y. Xiong, X. Cui, Attacks and countermeasures in internet of vehicles. Annals of Telecommunications 72(5-6)(2017) 283-295.

[36] Y. Takefuji, Connected Vehicle Security Vulnerabilities. IEEE Technology and Society Magazine 37(1)(2018) 15-18.

[37] C. Szilagyi, P. Koopman. Flexible Multicast Authentication for Time-Triggered Embedded Control Network Applications, in: Proc. the International Conference on Dependable Systems and Networks, 2009.

[38] S. Ezziri, O. Khadir, Amelioration of a proxy signature using RSA encryption, in: Proc. International Conference Networking, Information Systems and Security, 2019.

[39] M. Hu, Analysis and improvement of the security of DES algorithm. WIT Transactions on Information and Communication Technologies 57(2014) 317-324.

[40] G. De, A. Sison, R. Medina, MD5 secured cryptographic hash value, in: Proc. International Conference on Machine Learning and Machine intelligence, 2018.