Research of Authenticated Encryption Security Protocol for FlexRay In-vehicle Network

Meng-Zhuo Liu, Yi-Hu Xu, Yu-Jing Wu, and Yi-Nan Xu

***Abstract*—Since the appearance of internet of vehicle (IoV), the in-vehicles network is no longer an isolated system. The OBD-II hardware, GPS, and the WiFi system could cause intrusions during communications. There are many unresolved network security problems yet in in-vehicle bus protocol, like controller area network (CAN), local area network (LIN), and FlexRay. In this paper, having considered the vulnerabilities and a hidden attacking of FlexRay, an authenticated encryption protocol is designed based on the Advanced Encryption Standard (AES)-128 and SHA-1 algorithm for FlexRay. We apply the protocol into a simulated FlexRay bus system which is build by CAN software environment. The protocol is also applied in a real Flexray network build by MC9S12XF512 evaluation board The both experimental results show that the security and reliability of in-vehicle bus system are enhanced.**

***Index Terms*—FlexRay, in-vehicle bus network, network reliability, security protocol.**

I. Introduction

In the past tens of years, well developed traditional in-vehicle bus systems provide basic hardware supports. With beginning of the internal communication in vehicle, LIN, CAN, and FlexRay are used in the bus protocol to achieve the internal communication in vehicle. Based on the technology of x-by-wire and mechanotronics, the in-vehicle bus system is the main approach for ECU communications in the future motoring. In fact, provided convenient service combing with drivers other products like smart phone or Bluetooth are always faster than the security upgrade of systems. Thus, there are new challenging problems to be solved for sustainable development. One of problems is how to defend attack and patch the vulnerability on data and physical layers.

Recently, a lot of projects have carried out to improve the in-vehicle network. In the EVITA project [1], requires of the in-vehicle network are defined including information safety, the confidentiality, and concerned infrastructures etc. [2]. The project managed by Wolf stuff provides several of encryption algorithms to embed a security block in vehicle respectively [3]. In 2015, when the malicious APP on the mobile phone linking to the infortainment system, the security protocol is protected the system from this attack is by means of sending the authentication data using the extend ID and CRC field [4]. In 2017, considered of real time transmission, a light weight authentication protocol is proposed [5].

At the same time, new problems are updated too. For example, intruding into in-vehicle network through OBD-II interface could obtain and control messages including control the key functions and the engine. That is, the vehicle is totally controlled [6]. Intruder could gain control of vehicle telematic component by intruding the vehicle telephone [7]. The man-in-the-middle attack is also available to get the messages transmitted between vehicle and external environment [8]. In 2015, it is reported that a remote intrusion is possible in the distance of 16 kilometers [9]. Intruder could control the infortainment system, the steering system, and braking system.

In this paper, we propose an authenticated encryption protocol in where AES-128 algorithm is used in the data encryption. And HMAC (Hash-based message authentication code) based on SHA (secure hash algorithm) -1 is used in the authentication. It means that it is extremely hard to deduce original message through the digest. Through analyzing some representative attack model for in-vehicle bus system, the experimental result and performances will be discussed.

II. FlexRay Communication Protocol

FlexRay is a high speed network communication system that supports a time-triggered scheme and an optional event-triggered scheme. This FlexRay frame consists of three segments: header segment, payload segment, and trailer segment as depicted in Fig. 1. The first five bits define the basic features of the frame. Frame ID (11 bits) is defined as the slot position in the static segment and used to indicate the priority of the frame for the dynamic segment. Payload Length (7 bits) is defined as the data length. Header CRC (11 bits) is a Cyclic Redundancy Check, which is computed over the Sync Frame Indicator (1 bit), Startup Frame Indicator (1 bit), Frame ID (11 bits), and Payload Length (7 bits). Cycle Count (6 bits) is the serial number of the frame defined locally in the node. Trailer Segment (24 bits) is for Cyclic Redundancy Check, which is computed over the header segment and payload segment.

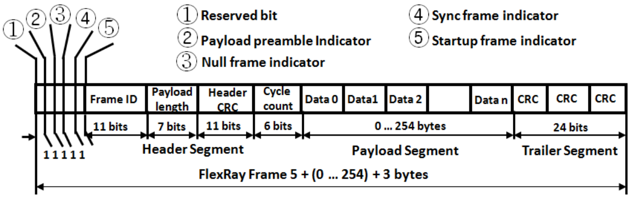


Fig. 1. FlexRay frame format.

Payload Segment (0 to 254 bytes) contains main data. These data will be transferred between vehicles, clouds, and

Manuscript received August 3, 2018; revised October 12, 2018. This work was supported by the National Natural Science Foundation of China (61763047).

The authors are with the College of Engineering of Yanbian University, Yanji, 133002, China (e-mail: ynxu@ybu.edu.cn).

DOI: 10.7763/IJCTE.2018.V10.1221

mobile terminals, connect the vehicle with the external environment. The previous study showed an efficient security protocol with respect to authentication delay and communication load. In this paper, based on these data messages, this concept will be used on FlexRay to control the braking system or start the car.

III. Attack Model

A. Malicious Messages Injection

Attack processes in this model are as follows: Steals the valid messages; Edits a malicious message with the format of a valid one; Eavesdrop the in-vehicle communication and inject malicious messages. Because there is no authentication mechanism, malicious messages are regarded as a trusted source like other ECU, and will be perform the corresponding operation.

B. Message Replay

For attacker, they do not need decrypt the ciphertext and understand the meaning of messages. The intruder can directly re-send the stolen messages in the bus system without authentication mechanism. Receiver ECU will accept the messages because they are valid. Besides, attacker can also inject meaningless messages to destroy the real-time transmission.

C. Message Modification

The intruder can modify the un-encrypted messages to confuse driver or give a malicious command to ECU. For example, the real speed may not equal to that display on the dashboard and etc. Attacker could still modify a CRC field to match original message.

IV. Security Protocol

A. The Design Goal

On the purpose of transmitting the FlexRay data frame efficiently and keeping in-vehicle network security, the security protocol need to achieve authentication, confidentiality, and real-time transmission objects.

Authentication: This object is to distinguish the received message from a re-sent malicious one. The object could be achieved through adding a special identifier into messages. In our proposed protocol, we use HMAC based on SHA-1 algorithm as an identifier. The receiver ECU can use this identifier to distinguish true messages from malicious ones.

Confidentiality: This is aim to defend from reading malicious message. Under the IoV background, it is easy to access the message shuttling vehicles and external environments. The concept used in this object is that the encrypt message could be understood in the case of getting the encryption key only. In our proposed protocol, AES-128 encryption algorithm is used to achieve confidentiality.

Real-time Transmission: Every in-vehicle bus protocol has a strong require of real-time transmission. Through adding a security mechanism, it will realize a message transmission without big delays. If the message did not arrive the target ECU in time, there is no meaning to keeping security.

B. AES-128 Encryption Algorithm

AES is a widely used symmetric encryption algorithm [10]. Before encryption process, the input will be copied into a 4 × 4 column-major order matrix of bytes, which is termed State. All the operation of AES is executed on this matrix.

$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	$S_{0,3}$
$S_{1,0}$	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$
$S_{2,0}$	$S_{2,1}$	$S_{2,2}$	$S_{2,3}$
$S_{3,0}$	$S_{3,1}$	$S_{3,2}$	$S_{3,3}$

Fig. 2. Matrix of state.

AES-128 is consists of 10 rounds of similar operation, and each round can be divided into four or three stages: They are substitute bytes, shift rows, mix columns, and add round key. Fig. 3 shows the encryption processing in detail.

Substitute bytes: There is S-box defined in AES-128, which is a 16 × 16 matrix containing 256 8-bits value. When substitute a certain byte of data in the State, the leftmost 4 bits are used as the column number, and the rightmost 4 bits are used as the row number. Then, based on column and row number, a new value is indexed from S-box to replace the corresponding byte in State.

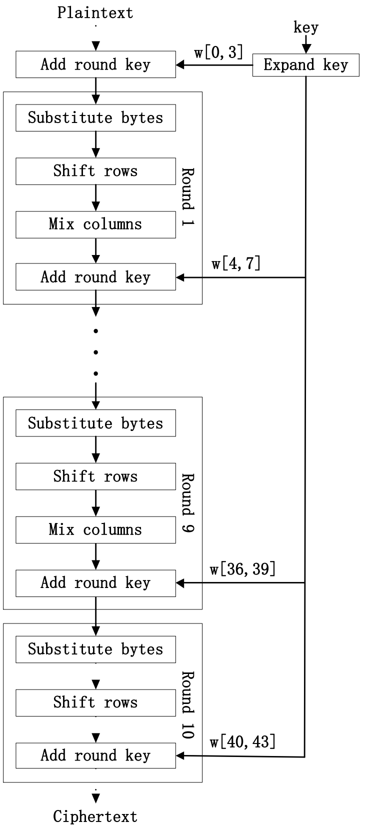


Fig. 3. AES-128 encryption processing.

Shiftrows: The first row keeps original sequence. All bytes from the second to the forth rows will be circularly shifted toward the left direction with a shift order of one, two, and three bytes from up to down.

Mixcolumns: Multiply State matrix with a specific matrix. Obtained new State will replace the original one.

Addroundkey: This stage will pick an expanded key corresponding to the round. Then, execute XOR operation between State and expanded key.

The key used in each round is the expanded original key. Expansion algorithm place the original key into the first 4 words of expand key array W. The rest of word *W[i]* in expand key will be calculated by *W[i-1]* and *W[i-4]*. When calculate the word with the index number of multiple of 4, there is a more complex operation.

Key expansion can defend cryptanalysis attack efficiently. And there is no chance for attacker to modify a message or send a malicious message without knowing the secret key.

C. HMAC Authentication Algorithm

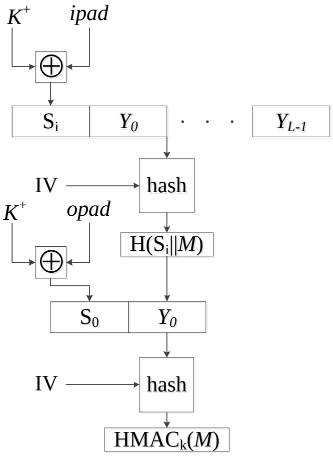


Fig. 4. HMAC processing.

Fig. 4 shows the HMAC algorithm in detail.

MAC is usually placed at the terminal of sent message. Using MAC, ECU could confirm the message as an unmodified true source. It is an efficient way that use hash function to calculate MAC. As well as, it is easy to replace the original one by a HMAC block when there is a more efficient hash algorithm.

In our proposed protocol, we use SHA-1(secure hash algorithm) in HMAC. The input length of SHA-1 is less than 2^{64} bits, and the output is a 160-bits digest. It is considerably difficult to deduce the original message through digest, and is impossible for attacker to generate correct HMAC without knowing secret key. Therefore, if the message replay attack takes place, receiver ECU can drop the message by compare HMAC.

The definition of HMAC can be given as

$$HMAC(K,m) = H((K^+ opad) \,||\, H((K^+ ipad) \,||\, m)) \tag{1}$$

where *H* represents hash function, *K* represents encryption key, K^+ represents the expanded key which add 0 at the left of original key, *m* represents the input of algorithm, and "||" denotes concatenation. The *ipad* and *opad* are two fixed constant. Their value are given as

ipad = the bytes 0x36 repeated 64 times
opad = the bytes 0x5c repeated 64 times

D. Design of Security Protocol

The proposed protocol apply AES algorithm in CTR (counter) mode. The operating process of CTR is shown in Fig. 5.

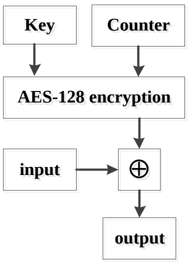


Fig. 5. AES-CTR mode.

In CTR mode, every read value is encrypted. The outputs comes from XOR operation on inputs and encrypted data. Because the character of CTR, there is no need of AES decryption algorithm. When encryption, the input is plaintext while the output is ciphertext. Conversely, when decryption, the role of input and output is exchanged.

In proposed protocol, we use AES-128 and HMAC to achieve the object of confidentiality and authentication respectively. Flowcharts of transmission and reception period are shown in Fig. 6 and Fig. 7, respectively. The variables used in flowcharts are explained in Table I.

TABLE I: THE VARIABLES USED IN FLOWCHARTS

variable	explanation
CTR_{cycle}	Counter value of cycle
CTR_m	Counter value of transmitted/received quantity for a certain message
CTR_{mac}	Counter value of transmitted/received quantity for HMAC of a certain message
C_{temp}	Encryption result of counter value
C_{send}	Ciphertext sent to receiver
M_{send}	HMAC sent
M_{rece}	HMAC received
EK	AES encryption key
AK	Authentication key of HMAC
P	plaintext

In the proposed protocol here, encryption keys are initialized before starting ECU communication. Because of symmetric encryption algorithm of AES, encryption keys have same values at the sending and the receiving terminals. And, encryption keys of AES and HMAC in both terminals are regenerated with a communication cycle.

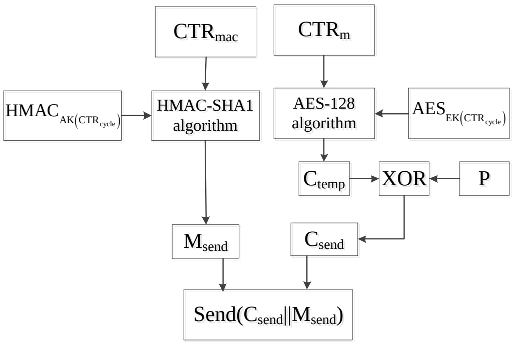


Fig. 6. Flowchart of transmission period.

As we can see from Fig. 6, before sending a message, the sender ECU will obtain CTR_{cycle}, CTR_m, and CTR_{mac} firstly. Next, using CTR_{cycle} as an index number, get corresponding encryption keys of $EK(CTR_{cycle})$ and $AK(CTR_{cycle})$. Then, use $EK(CTR_{cycle})$ to encrypt CTR_m with AES mode, obtain the output result of C_{temp}. Finally, through XOR operation between C_{temp} and P, C_{send} will be calculated.

After encryption processing, M_{send} will be calculated using AK (CTR_{cycle}) and CTR_{mac}, where CTR_{mac} is the input of HMAC algorithm. When all calculations are done, M_{send} will be placed at the terminal of C_{send} waiting to be sent. After sending message, CTR_{mac} and CTR_m will be increased by ones respectively to ensure value synchronization in both of sending and receiving terminals.

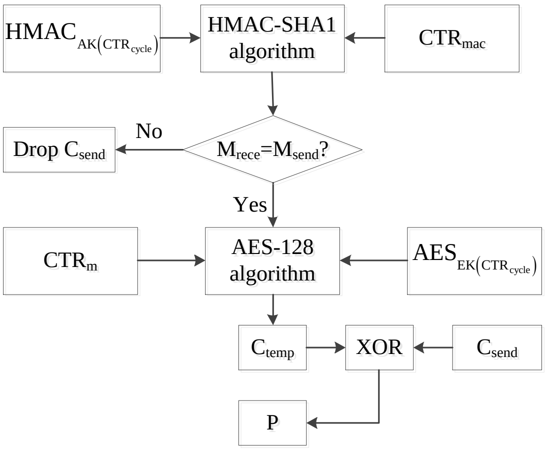


Fig. 7. Flowchart of reception period.

Fig. 7 shows the receiving process of ECU. When receive a message, ECU will obtains CTR_{cycle}, CTR_m, and CTR_{mac} firstly. Then, AK (CTR_{cycle}) is use to calculate HMAC of CTR_{mac} to obtain a message authentication of M_{rece}. Only when M_{rece} is equal to M_{send}, receiver ECU will accepts C_{send}. This process realized an authentication achievement. Finally, EK (CTR_{cycle}) is used to encrypt CTR_m with AES mode, and plaintext can be obtained through XOR operation between C_{temp} and C_{send}.

The ciphertext and HMAC are placed in the payload segment of FlexRay. The format of them is showed in Fig. 8.

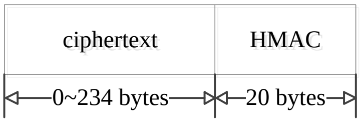


Fig. 8. Message format.

E. Security Analysis

Suppose that the attacker don't know the encryption and the authentication keys. Attacker could not generate a valid HMAC. Receiver ECU will drop this wrong HMAC, the malicious message. In the case of message attack replays, because the HMAC are regenerated with a communication cycle, malicious messages will not match to the current value of receiver ECU and be dropped. By the adoption of AES-CTR mode, the counter value will always keep fresh after sending. Therefore, the system defend from the attack of fuzzing test.

V. Experimental Result

We use the CANoe software system of Vector company to build a simulated FlexRay network with two virtual ECU(shown in Fig. 9), where ECU_1 is send only and ECU_2 is receive only. We implement our proposed security protocol and build a DLL to apply it into the network.

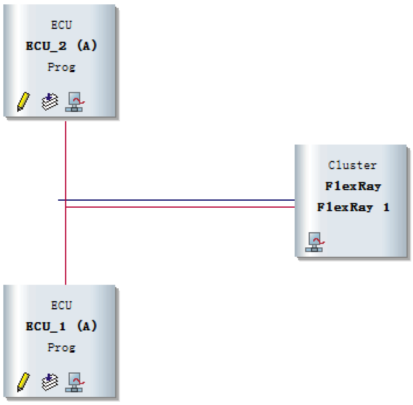
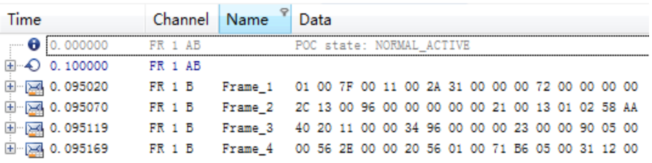


Fig. 9. Simulated network structure.

There are four messages transmitted on the network, before a security protocol processing, the state of transmission is shown in Fig. 10.



```
Time        Channel  Name     Data
0.000000    FR 1 AB           POC state: NORMAL_ACTIVE
0.100000    FR 1 AB
0.095020    FR 1 B   Frame_1  01 00 7F 00 11 00 2A 31 00 00 00 72 00 00 00 00
0.095070    FR 1 B   Frame_2  2C 13 00 96 00 00 00 00 00 21 00 13 01 02 58 AA
0.095119    FR 1 B   Frame_3  40 20 11 00 00 34 96 00 00 00 23 00 00 90 05 00
0.095169    FR 1 B   Frame_4  00 56 2E 00 00 20 56 01 00 71 B6 05 00 31 12 00
```

Fig. 10. Transmission state without security processing.

Then, we use the proposed protocol to process the message. By the end of system initialization, there is no message transmitted on the network, so the counter value of every ECU is zero. When ECU_1 is about to send Frame_1, the message will be processed firstly by security protocol. The calculated result is shown in Table II.

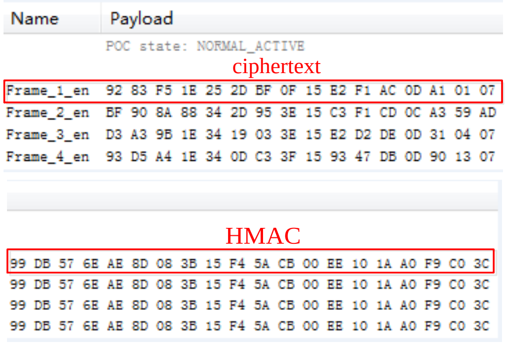
The transmission state of processed messages is shown in Fig. 11. As we can see from Fig. 11, the encrypted data and HMAC could be transmitted in real-time. And the data is as we expect in Table II.

TABLE II: PROCESSED DATA OF FRAME 1

variable	data
plaintext	01007F00 11002A31 00000072 00000000
Counter value	0
Encryption key	1b200b19 7e5ef522 05465540 1716483a
HMAC key	60814fdc 222a9088 46eeb814 de5e0bdb
ciphertext	9283f51e 252dbf0f 15e2f1ac 0da10107
HMAC	99db576e ae8d083b 15f45acb 00ee101a a0f9c03c

On the purpose of further analyzing this proposed protocol, we build a two-node FlexRay network. The Freescale MC9S12XF512 evaluation board (EVB) is used to setup the actual test environment.

The MC9S12XF512 MCU has a build-in FlexRay module supporting bitrate up to 10 Mbps. The module is constructed according to the FlexRay Communication System Protocol Specification V2.1 [11].



```
Name        Payload
            POC state: NORMAL_ACTIVE
                         ciphertext
Frame_1_en  92 83 F5 1E 25 2D BF 0F 15 E2 F1 AC 0D A1 01 07
Frame_2_en  BF 90 8A 88 34 2D 95 3E 15 C3 F1 CD 0C A3 59 AD
Frame_3_en  D3 A3 9B 1E 34 19 03 3E 15 E2 D2 DE 0D 31 04 07
Frame_4_en  93 D5 A4 1E 34 0D C3 3F 15 93 47 DB 0D 90 13 07

                          HMAC
99 DB 57 6E AE 8D 08 3B 15 F4 5A CB 00 EE 10 1A A0 F9 C0 3C
99 DB 57 6E AE 8D 08 3B 15 F4 5A CB 00 EE 10 1A A0 F9 C0 3C
99 DB 57 6E AE 8D 08 3B 15 F4 5A CB 00 EE 10 1A A0 F9 C0 3C
99 DB 57 6E AE 8D 08 3B 15 F4 5A CB 00 EE 10 1A A0 F9 C0 3C
```

Fig. 11. Transmission state of processed frame.

The proposed security protocol is developed using C code by CodeWarrior Development Studio. And the BitRate is set to 10Mbps by configuration of the hardware parameters. The block diagram of the test environment is shown in Fig. 12. And the actual test environment of FlexRay network is given at Fig. 13.

As shown in Fig. 13, the experimental result was displayed on the LCD screen which is produced by CANoe software. These messages can be transmitted in real-time on the bus system. It indicates that the proposed protocol can be used in FlexRay network with few transmission delay.

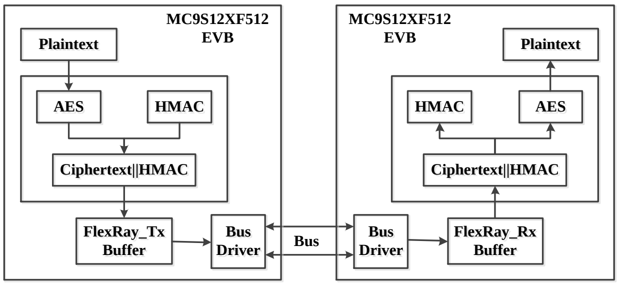
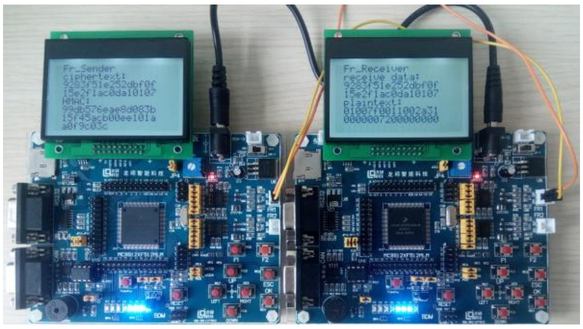


Fig. 12. Block diagram of test environment.



```
Fr_Sender
ciphertext:
9283f51e252dbf0f
15e2f1ac0da10107
HMAC:
99db576eae8d083b
15f45acb00ee101a
a0f9c03c
```

```
Fr_Receiver
receive data:
9283f51e252dbf0f
15e2f1ac0da10107
plaintext:
01007f0011002a31
0000007200000000
```

Fig. 13. Actual test environment of FlexRay network.

VI. Conclusion

The in-vehicle network is no longer a close system and securities begin to be considered seriously. A security protocol proposed here is simulated FlexRay network and build by CANoe software. Through configuration of this proposed protocol into MC9S12XF512 evaluation board, the experimental result shows that protocol is suitable for FlexRay network.

References

[1] The EVITA Project. (2008). [Online]. Available: https://www.evita-project.org

[2] A. Ludovic, E. K. Rachid, H. Olaf, R. Yves, S. Hendrik, S. Herve, W. Benjamin, and W. Markoet, "Secure automotive on-board electronics network architecture," presented at FISITA 2010 World Automotive Congress, Budapest, Hungary, May 30-June 4, 2010.

[3] W. Marko, A. Weimerskirch, and T. Wollinger, "State of the art: Embedding security in vehicles," *EURASIP Journal on Embedded Systems*, vol. 2007, no. 1, pp. 1-16, Dec. 2007.

[4] W. Samuel, H. J. Jo, and H. L. Dong, "A practical wireless attack on the connected car and security protocol for in-vehicle can," *IEEE Trans. Intelligent Transportation Systems*, vol. 16, pp. 993-1006.

[5] P. Mundhenk, A. Paverd, A. Mrowca, S. Steinhorst, M. Lukasiewycz, S. A. Fahmy, and S. Chakraborty, "Security in automotive networks: lightweight authentication and authorization," *ACM Trans. Design Automation of Electronic Systems*, vol. 22, no. 2, pp. 1-27, Mar. 2017.

[6] K. Koscher, A. Czeskis, F. Roesner *et al.*, "Experimental security analysis of a modern automobile," *2010 IEEE Symposium on Security and Privacy*, pp. 447-462, 2010.

[7] S. Checkoway, D. McCoy, B. Kantor *et al.*, "Comprehensive experimental analyses of automotive attack surfaces," in *Proc. 20th USENIX Conference on Security*, pp. 6-6, 2011.

[8] P. Jonathan and S. E. Shladover, "Potential cyberattacks on automated vehicles," *IEEE Trans. Intelligent Transportation Systems*, vol. 16, no. 2, pp. 546-556, Sep. 2014.

[9] C. Miller and C. Valasek, "Remote exploitation of an unaltered passenger vehicle," presented at Black Hat, Las Vegas, USA, August 1-6, 2015.

[10] *Advanced Encryption Standard*, NIST FIPS-197, 2001.

[11] FlexRay Consortium *et al.*, "FlexRay communications system-protocol specification," *Version*, vol. 2, no. 1, 2005.

Meng-Zhuo Liu was born at Jilin Province, China. He received the bachelor degree in communication engineering from YanBian University, China, in 2017.

He is a currently working toward a master degree in the area of in-vehicle network, which include the design of security architecture of FlexRay.

Yi-Hu Xu was born at Jilin Province, China. He received the Ph.D. degree in electronics engineering from the Chonbuk National University, Korea, in 2014.

He is a lecturer of the division of electronic and communication engineering of Yanbian University, Yanji, China. His research interests include the automobile electronic control and network.

Yu-Jing Wu was born at Jilin Province, China. She received her M.S. and Ph.d in electronic and information engineering from Chonbuk National University, South Korea, in 2013 and 2016, respectively.

She is a lecturer of the division of electronic and communication engineering of Yanbian University, China. Her research interests are in the area of VLSI implmentation for digital signal processing and communicaiton system, which include the design and in implementation of security protocol for in-vehicle networks.

Yi-Nan Xu was born at Jilin Province, China. He received the Ph.D. degree in electronics engineering from the Chonbuk National University, Korea, in 2009.

He is a professor of the division of electronics and communication engineering of Yanbian University, Yanji, China. His research interests include the in-vehicle network and automobile electronic control.