Security Authentication System for In-Vehicle Network

Hiroshi UEDA*, Ryo KURACHI*, Hiroaki TAKADA, Tomohiro MIZUTANI, Masayuki INOUE and Satoshi HORIHATA

One of the main concerns for the security of in-vehicle data is spoofing messages on the in-vehicle network. Controller Area Network (CAN) is the most extensively embedded network protocol in vehicles. In the last decade, security attacks in vehicles have been increasing and have been reported in several papers. Therefore, security measures are expected that meet the requirements of real time and cost constraint for in-vehicle control network. In this paper, we propose centralized authentication system in CAN with improved CAN controller. Our experimental results demonstrate that our proposal method is effective on real in-vehicle network environments.

Keywords: embedded security, in-vehicle control network, Controller Area Network (CAN)

1. Introduction

An automobile produced today may be equipped with more than 70 electronic control units (ECUs)[1] that are controlled via a controller area network (CAN),[2] local interconnect network (LIN),[3] FlexRay,[4] or other suitable network. While CAN is the most popular protocol among in-vehicle control networks and is used in most of the vehicles on the market, the vulnerability of this protocol to security threats has been pointed out. Falsification of meter readings, disablement of brake function, and other unauthorized control by a spoofed message injected into the network have been demonstrated as case examples of attacks on CAN.[5] In addition, the maximum data transfer rate of CAN is 1 Mbps and the maximum payload of a message is 8 bytes. Due to these properties, it is difficult for CAN to directly use the security measures that have been developed for consumer products.

In this paper, the authors propose a centralized CAN security monitoring system comprising an improved CAN controller.*1 In a timing verification test of a prototype substrate comprising a field-programmable gate array (FPGA), we demonstrated that the system proposed here is practicable.

2. Security Risk of CAN

2-1 Features of CAN

CAN is a communication protocol widely used for in-vehicle control systems. This protocol was standardized by ISO 11898 and ISO 11519 with a focus on the first and second layers of the OSI reference model. The following are the principal features of CAN:

(1) Bus topology
CAN is widely used in a bus topology that two or more ECUs are connected to a communication line.

(2) Multi-master
Since each node can immediately transmit a message on a CAN bus as needed, it is easy to add CAN messages and nodes.

(3) Arbitration of transmission right
When two or more nodes transmit messages on a CAN bus at the same time, the transmission right is arbitrated based on CAN-ID. After the arbitration, the CAN message containing the message with the highest priority is transmitted first. As a result, transmission of lower-priority messages is delayed until higher priority messages are sent out.

2-2 Security risk of CAN

From the attack cases[5] experienced in the past, the following two use cases can be supposed as possible spoofing attacks on CAN.

Use case 1: Unauthorized alteration of ECU software

Figure 1 shows an example of spoofed message transmission by an ECU after its authorized program is replaced by a malicious program.

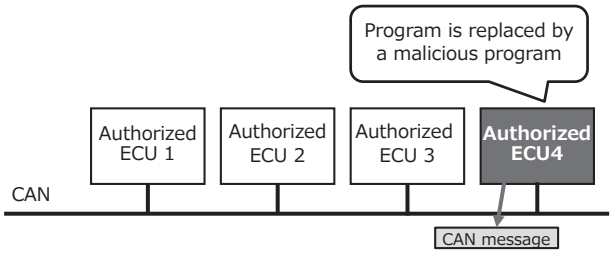


Fig. 1. Replacement of authorized ECU program by malicious program

Use case 2: Connection of unauthorized device

Figure 2 shows an example of spoofed message transmission by an unauthorized device connected on a CAN bus.

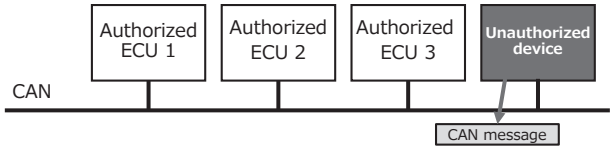


Fig. 2. Connection of unauthorized device on CAN bus

3. Security Measure for CAN

3-1 Conventional research

In the recent situation in which many case examples of attacks on in-vehicle control systems are reported, a lot of research has been conducted on developing effective measures for protecting the control systems from attacks.[6]-[10] However, these measures have the same problem that all nodes on the network must implement the measures.

3-2 The method proposed in this paper: security monitoring system

The security monitoring that we propose in this paper is schematically illustrated in **Figure 3**. The CAN protocol is vulnerable to spoofing attacks since it has no authentication system as described before. The monitoring system proposed here uses message authentication code (MAC)*2 that is the generally effective against spoofing. The objective of the proposed security monitoring system is to prepare against spoofing attacks. In particular, a monitoring node authenticates each ECU and verifies the message authentication codes assigned to the CAN messages. It is essential to install a special purpose CAN controller in the monitoring node.

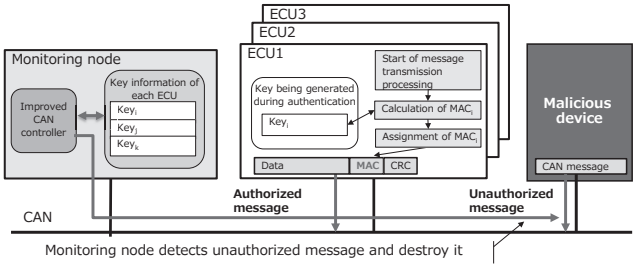


Fig. 3. Proposed security monitoring system

The special purpose CAN controller uses an error frame to overwrite spoofed messages on a real time basis. The advantage of the proposed security monitoring system is that the only new hardware required is the monitoring node on the CAN bus. For other ECUs, only the software for node authentication and key exchange needs to be modified.

A transmission node (authorized ECU) calculates the MAC using an encryption key and gives a part of the calculated MAC to the payload and transmits the data frame. The monitoring node refers to the CRC, *3 and then verifies the MAC using the same encryption key as that used for each transmission node. If the new CAN controller detects a spoofed message, the controller overwrites the message using an error frame. The security monitoring system proposed here consists of the following two phases:

(1) Phase of node authentication and key delivery
(2) Phase of monitoring spoofed message and overwriting the message with error frame on a real time basis

In the systems that have conventionally been researched, the key delivery mechanisms are very complicated and impose a large communication load onto the CAN buses. Therefore, these systems are unacceptable for in-vehicle control systems that require real-time data processing. For vehicle applications, simple authentication and key delivery protocol are essential to reduce calculation time.

3-3 Mutual authentication of node and key exchange

Since our security monitoring system does not encrypt the payload of the data frame, it is unnecessary for all nodes to have an identical encryption key. Therefore, we determined to use a challenge response system for authentication between the monitoring node and each transmission node (ECU). To describe it more concretely, we use the following mutual authentication sequence (**Figure 4**).

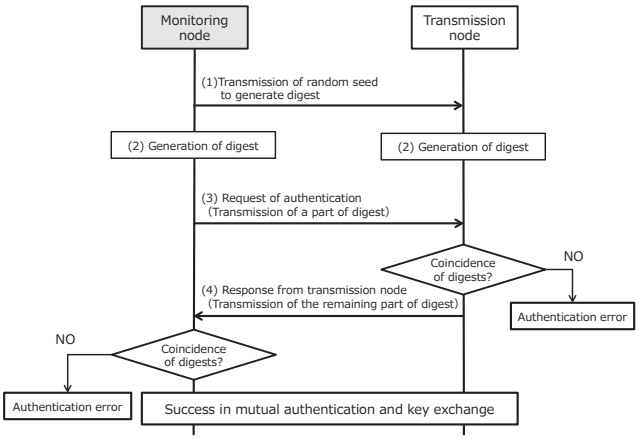


Fig. 4. Protocol for mutual authentication and key exchange

(1) The monitoring node transmits a random seed to the transmission node.
(2) After the transmission node receives the random seed from the monitoring node, both nodes calculate the digest. A hash function*4 is usually used for the digest calculation.
(3) The monitoring node transmits a part of the digest to the transmission node.

(4) The transmission node calculates the digest and compares it with the part of the digest transmitted from the monitoring node. When the two digests are equal to each other, the transmission node transmits the subsequent digests to the monitoring node.
(5) The monitoring node conducts authentication by calculating the digest and comparing it with the part of the digest received from the transmission node.

3-4 Pre-shared information and encryption key

The security monitoring system we propose here assumes that no tamper-resistant*5 memory is installed in the transmission node. Under the above assumption, we use a program code and ROM information containing a unique ID as a pre-shared key. This means that the monitoring node is required to have all transmission node programs. However, in practice, the monitoring node can reduce memory usage by having several kinds of preliminarily calculated authentication codes. We used the SHA-256 hash function to demonstrate our security monitoring system. However, many other methods can be used for our system. The details of the encryption key are described below.

(1) Pre-shared information

We used a program code as pre-shared information to simplify the demonstration. In this case, authentication code generation time depended on program size. Since the leakage of program code is a shortcoming of our system, the effects of leakage must be minimized. In an actual use environment, it is preferable for all ECUs to individually have a key as shared information.

Generation of an authentication code is expressed by the following formula:

$$AUTHKEY_I = SHA256\ (MSG \parallel NONCE)$$

$AUTHKEY_I$ represents the authentication code of transmission node$_I$ and the *SHA256* function represents a function for SHA-256 hash calculation. *MSG* represents the program code of the transmission node and *NONCE* represents a random seed.

(2) MAC generation/verification key

In the security monitoring system proposed in this paper, the remaining part of the above-described $AUTHKEY_I$ is used as the MAC generation/verification key. In consequence, a 128 bit MAC generation key was mounted.

3-5 Authentication message

The security monitoring system proposed here separates the message transmission frame from the authentication information transmission frame. The difference between the CAN message used for our system and a generally used CAN message is shown in **Figure 5**. For the CAN message in our system, a part of the MAC is given to a part of the payload. However, the payload is not encrypted. In the mutual authentication shown in **Fig. 4**, a part of the generated digest is given to the whole payload.

3-6 Operation of monitoring node

In a generally used CAN protocol, all reception nodes use a CAN controller to perform CRC check and thus to detect transmission errors. In our system, the MAC is given to a part of the payload to assure the integrity of the CAN message, as shown in **Figure 6**.

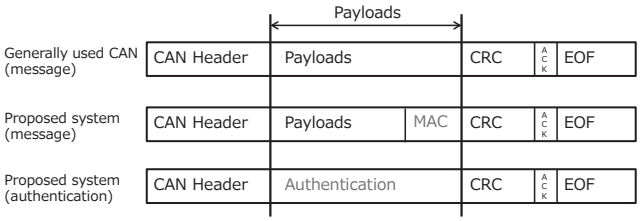


Fig. 5. CAN protocol of proposed security monitoring system

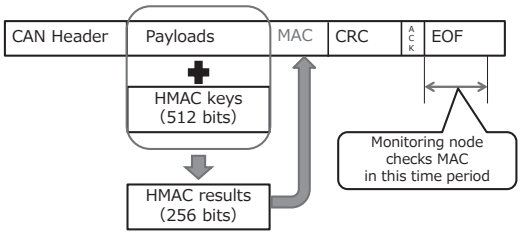


Fig. 6. HMAC calculation timing

The role of the MAC is to protect data from spoofing on the in-vehicle network. For the purpose of maintaining interchangeability with existing CAN controllers, our system does not replace CRC with the MAC.

When the monitoring node receives a message, the node checks the CAN-ID to judge whether or not the MAC has been given to the message. When the MAC has been given to the message, the monitoring node immediately calculates the HMAC to verify the MAC. At this point of time, no node has transmitted an ACK signal to the CAN bus. However, if the monitoring node detects the MAC error, the node overwrites the spoofed message with the error frame up to the end of frame (EOF). Thus, substitutional verification of the MAC by the monitoring node makes it possible for our system to obstruct the transmission of spoofed messages without verifying the MAC by all reception nodes.

Our system is also able to prevent replay attacks by giving a part of monotonic counter*6 to the payload.

3-7 HMAC algorithm

The security monitoring system we propose here uses the following HMAC function:

$$CalcMAC_i = HMAC\ (ID_i, D_i, FC_i, Key_I)$$

The HMAC function is a hash function resembling HMAC-SHA256. ID_i represents a CAN-ID. D_i represents a part of a payload after MAC_i and LC_i are removed. FC_i represents a complete monotonic counter for message *i*. Key_I represents the encryption key ($AUTHKEY_I$) for the transmission node$_I$.

4. Security and Performance Analysis

4-1 Integrity of message

The security strength of a HMAC increases as the length of the MAC increases. For the protocols of PCs and other devices that use SHA-256, the length of the MAC is 32 bytes. However, since the maximum payload of the CAN is 8 bytes, it is impossible to accommodate all MACs in a single frame. We determined to set the length of the MAC at 1 byte.

For a MAC of 1 byte in length, the total number of necessary attacks on a particular message is 2^8. Therefore, the probability of successful spoofing by random attack is $1/2^8$, which means that 2^8 messages must be transmitted until the attack succeeds. Though this figure is not enough to perfectly guarantee the security of the CAN, rotating the keys after transmission of several messages and other additional preventive measures will make it more difficult to attack the CAN by spoofing.

4-2 Real-time limitation

Real-time limitation has no direct relationship with security but is important for in-vehicle control systems.

(1) Number of messages necessary for authentication and key exchange

As the number of nodes increases by one, the number of messages necessary for authentication and key exchange increases by two. One is the message transmitted from the monitoring node (request for authentication in **Fig. 4**), and the other is the message transmitted from the added node (authentication response in **Fig. 4**). As described above, the number of messages differs depending on the number of nodes to be authenticated. The number of messages can be calculated from $2 \times n$ (n: the number of nodes to be authenticated).

(2) Overhead by MAC

The system design engineer in charge can determine the length of MAC. We conducted calculations on the assumption that the length of MAC would be 1 byte. Since the length of the counter for preventing replay attacks is 4 bits, an overhead with a total of 12 bits is created. This overhead accounts for 19% of the total payloads and falls within the range of the extension identifier (18 bits) of the frame. Therefore, this overhead is considered to be fully acceptable for actual vehicles.

5. Mounting Evaluation Results for Proposed Security Monitoring System

To demonstrate the above-described protocol and system, we used an Altera FPGA development board and CAN transceiver board. The CAN controller with built-in HMAC, which is the most outstanding feature of our system, was mounted on an FPGA.

The FPGA development board used for the evaluation (DE2-115 development and education board) comprised a 512-kbyte FLASH memory and 20-kbyte RAM. The actual evaluation environment is shown in **Photo 1**.

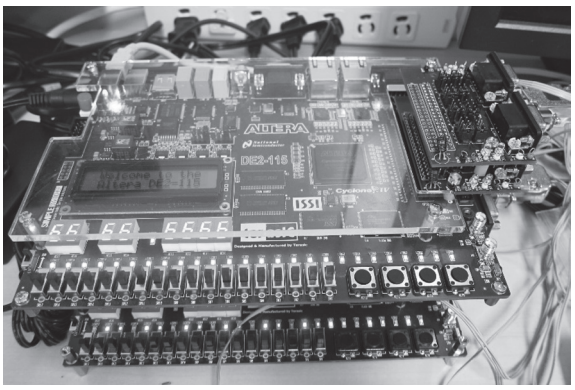


Photo 1. Evaluation environment

In this evaluation environment, the time necessary to inspect a MAC with a single reception frame was measured to be approximately 2.12 μs. When the transfer rate of the CAN is assumed to be 1 Mbps, the inspection time falls within 3 bit time. In other words, an error frame can be transmitted within 9 bit time (the sum of ACK (2 bit time) and EOF (7 bit time)), which is the time necessary to calculate the CRC of the CAN message, inspect the MAC, and overwrite the MAC with the error frame. As described above, it was confirmed from the mounting evaluation that the security monitoring system proposed here has a sufficiently feasible performance.

6. Conclusions

Attacks on in-vehicle control units via in-vehicle control networks have been reported recently and various measures for protecting the networks from attacks have been proposed. However, most of these measures must be implemented for every node. In this paper, we proposed a centralized security monitoring system and discussed its demonstration results.

In future, we will demonstrate the effectiveness of key exchange to verify that our system works even in environments closer to actual in-vehicle environments.

Technical Terms

∗1 CAN controller: A controller that realizes the function of a CAN protocol.

∗2 Message Authentication Code (MAC): An authentication technique for confirming that the messages from a communication partner have not been tampered. This technique is used to verify and authenticate the integrity of messages.

∗3 CRC: Standing for cyclic redundancy check, CRC is a technique used mainly to detect data transfer errors. Since there is always the possibility that the same numerical values are output from different data, it is inappropriate to use output as a substitute for a hash value.

∗4 Hash function: A function for compressing a string of characters to certain length of data. The values calculated from this function are called "hash values" or simply "hash." SHA-1 and SHA-256 are typical hash functions. Since both of the functions are one-way functions, it is impossible to determine the original sentences from the generated data.

∗5 Tamper resistance: Difficulty of analyzing internal structures and stored data. The capability to prevent reading of confidential data by unauthorized means is called "tamper resistance."

∗6 Monotonic counter: A counter that monotonically increases the reading.

References

(1) J. Leohold, Communication Requirements for Automotive Systems, 5th IEEE Workshop Factory Communication Systems (2004)

(2) International Organization for Standardization, Road vehicles - Controller area network (CAN) - Part 1: Data link layer and physical signaling, ISO11898-1 (2003)

(3) International Organization for Standardization, Road vehicles - Local Interconnect Network (LIN) - Part 1: General information and use case definition, ISO/DIS 17987-1

(4) International Organization for Standardization, Road vehicles - Communication on FlexRay - Part 1: General information and use case definition, ISO10681-1 (2010)

(5) K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, Experimental Security Analysis of a Modern Automobile, IEEE Symposium on Security and Privacy (2010)

(6) A. V .Herrewege, D. Singelee, I. Verbauwhede, CANAuth - A Simple, Backward Compatible Broadcast Authentication Protocol for CAN bus, Embedded Security in Cars 9th, Dresden, Germany (Nov. 2011)

(7) A. Hazem, Hossam. A. H. Fahm, LCAP - A Lightweight CAN Authentication Protocol for Security In-Vehicle-Networks, Embedded Security in Cars 10th, Berlin, Germany (Nov. 2012)

(8) O. Hartkopp, C. Reuber, R. Schilling, MaCAN - Message Authenticated CAN, Embedded Security in Cars 10th, Berlin, Germany (Nov. 2012)

(9) T. Hoppe, S. Kiltz, J. Dittmann, Security threats to automotive CAN networks - Practical examples and selected short-term countermeasures, Proceedings of the 27th international conference on Computer Safety, Reliability, and Security, SAFECOMP '08, pp. 235-248 (2009)

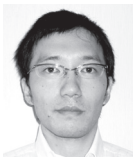
(10) T. Matsumoto, M. Hata, M. Tanabe, K. Yoshioka, A Method of Preventing Unauthorized Data Transmission in Controller Area Network, Vehicular Technology Conference (VTC Spring), 2012 IEEE 75th (2012)

Contributors (The lead author is indicated by an asterisk (*).)

H. UEDA*

• Manager, Information Network R&D Division, AutoNetworks Technologies, Ltd.

R. KURACHI*



• Ph.D.
Designated Assistant Professor, Center for Embedded Computing Systems,Graduate School of Information Science, Nagoya University

H. TAKADA

• Ph.D.
Executive Director, Center for Embedded Computing Systems, Graduate School of Information Science, Nagoya University

T. MIZUTANI

• Information Network R&D Division, AutoNetworks Technologies, Ltd.

M. INOUE

• Assistant Senior Manager, Information Network R&D Division, AutoNetworks Technologies, Ltd.

S. HORIHATA

• Senior Manager, Information Network R&D Division, AutoNetworks Technologies, Ltd.