Spoofing Attack Using Bus-off Attacks against a Specific ECU of the CAN Bus

Kazuki Iehira*, Hiroyuki Inoue†‡, and Kenji Ishida†
*Faculty of Information Sciences,
Hiroshima City University, Hiroshima, Japan
Email: {iehira@net.info., hinoue@}hiroshima-cu.ac.jp

†Graduate School of Information Sciences,
Hiroshima City University, Hiroshima, Japan
‡Connected Consumer Device Security Council (CCDS),
Okinawa, Japan

***Abstract*—This paper reports spoofing attacks that exploit a vulnerability of the controller area network (CAN) protocol, which is often used in in-vehicle networks. However, authorized electronic control units (ECU) should be able to detect anomalous CAN message traffic. We focused on ECU states used for error handing. Bus-off attacks against the ECU have been proposed to induce transmission errors, and then transit the state of the target ECU to the bus-off state, in which the ECU cannot access the CAN bus. The attack combining the bus-off attack with the spoofing attack could not be detected by the authorized ECUs, and would consequently be a potential threat for the vehicle. In this paper, we propose a spoofing attack method that uses a bus-off attack and is not detected by the authorized ECUs. Based on the proposal, we implemented an attacker prototype using a field-programmable gate array. In a laboratory setting, we verified the attack in a simulated environment consisting of the attack hardware and ECU, and evaluated the effect of spoofing and the behavior of an actual car. The results showed that the transmission of regular messages was completely prevented, and the percentage of spoofing messages could be made 100%; that is, no error was detected in the ECUs of the car. We have verified the feasibility of the proposed method and the potential threat for actual cars. In the future, we will conduct studies to prevent vehicles from such attacks.**

Keywords—CAN, in-vehicle network, spoofing attack, bus-off attack

I. INTRODUCTION

The controller area network (CAN) protocol [1] is a protocol for in-vehicle networks and has a bus-type network topology. The mechanism of message encryption is not standardized. Spoofing attacks exploiting the vulnerability of the CAN protocol, which is often used for in-vehicle networks, are reported. As an example of a CAN attack, spoofing a message injection has been reported to make it possible to spoof a meter and control a brake [2][3][4]. In October 2016, a bus-off attack was reported as a new denial of service (DoS) attack against CAN [5]. In this attack, a transmission error is induced, and the target electronic control unit (ECU) is transited to the bus-off state, so it leaves the CAN bus. Furthermore, a stuff error can be intentionally caused to realize bus-off attacks without time restrictions such as the transmission start timing and cycle [6].

Theoretically, it is difficult for the receiving ECU to distinguish a regular CAN message from a message injected with a simple spoofing attack. However, it is possible to detect anomalies according to the cycle and content [7]. Therefore, if a spoofing message can be injected without being detected by an authorized ECU, a more effective spoofing attack becomes possible. In this research, we focused on an attacker blocking the transmission/reception of an authorized transmission ECU and transmitting a spoofing message. We propose a spoofing attack method where the receiving and transmission ECUs cannot detect anomalies. We created a prototype of the attack equipment and applied it to a simulated environment and vehicle CAN bus for evaluation and discussion.

II. RELATED WORK

A. Characteristics of the CAN Protocol

In the CAN protocol, a state representing the logic "1" is recessive, and a state representing the logic "0" is dominant. During simultaneous transmission of dominant and recessive bits, the resulting bus value will be dominant. In addition, bit-stuffing rules are applied to prevent synchronization from being lost by transmitting the same logic bits continuously. Therefore, when bits of the same logic continue for 5 bits, 1 logic-inverted bit is inserted. The CAN message does not have a source address and has only the CAN ID as a destination address. Furthermore, the CAN message is broadcast on the bus, so the receiving ECU cannot know the source. Therefore, spoofing attacks are easily possible.

In CAN, a message that transmits data is called a data frame, which is composed of the CAN ID, date length code (DLC), and data part. A message that announces an error is called an error frame, which is composed of a 6-bit error flag and 8-bit recessive. There are several kinds of errors. A bit error occurs when the transmitted bit logic differs from the received bit logic. A stuff error occurs when the bit-stuffing rule is violated. To control errors, when the ECU detects an error, it transmits an error frame to inform the other ECUs. When an error is detected in the CAN controller, the values of the transmit error counter (TEC) and receive error counter (REC) are increased. If the transmission ECU detects a bit error or a stuff error, the TEC of the ECU increases by 8. As shown in Fig. 1, the CAN controller has three states. The state transits

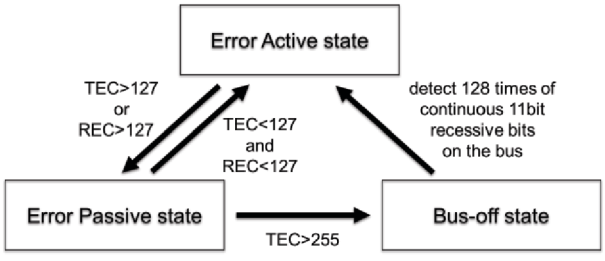


Fig. 1 ECU state transition

978-1-5386-4790-5/18/$31.00 ©2018 IEEE

according to the values of the TEC and REC, and access to the bus is restricted. The level of the error flag also changes. Table 1 shows the access restrictions and error flags transmitted according to the state. The error active state is normal operation, and there is no limitation on transmission. The error passive state is prone to error. The transmission waiting time during continuous transmission is set longer than usual, so as to not interfere with communication by other ECUs. Because only recessive bits are transmitted even when an error is detected, it cannot hinder the transmission of other ECUs. In the bus-off state, the ECU cannot participate in communication on the bus. Therefore, all transmissions and receptions are forbidden.

B. *Spoofing Attack on a CAN*

Attention to the information security of in-vehicle LANs and ECUs is increasing. In conventional spoofing attacks as mentioned above, regular messages are also transmitted, so complete impersonation is not possible. We propose a spoofing attack method where the receiving and transmission ECUs cannot detect anomalies. In the proposed method, the message transmitted by the authorized transmission ECU is intentionally set as an error. As a result, the state is transited to the bus-off state, and transmission/reception by the authorized transmission ECU is prevented. In addition, a spoof message is injected into the CAN bus.

C. *Bus-off Attack against an ECU*

1) *Bus-off Attack Using the Bit Error*

Here, we introduce a bus-off attack that uses a bit error to transit the target ECU to the bus-off state, as shown in Fig. 2. By inducing a bit error in the target ECU, the TEC of the target ECU is increased. Next, the attack procedure is shown. Just before the target ECU starts transmission, a meaningless message is injected. As a result, the transmission of the target ECU is blocked and stored in the buffer. At the same time, messages that satisfy the following conditions are stored in the buffer of the attacker ECU:

- It must have the same CAN ID as the target ECU.
- When the target ECU sends a recessive bit, the attacker sends dominant bits.

As a result, as shown in Fig. 2, the target ECU and attacker transmit at the same time. When the transmission bits are different, the TEC of these ECUs is increased by 8. By repeating this process, TEC > 255 is set, and the target ECU is transited to the bus-off state.

In this attack, when the attacker ECU and target ECU are in the error active state, the retransmission is also performed at the same time. Thus, the TEC can be increased particularly effectively. The increase in TEC at this time is obtained as follows:

$$\textbf{TEC of target ECU} = 8 \times \left\lceil \frac{128 - \max(TEC\ of\ Attacker, TEC\ of\ Target)}{8} \right\rceil \quad (1)$$

$$\textbf{TEC of attacker ECU} = \text{TEC of Target} - 8 \quad (2)$$

Note that in this attack the TEC of the attacker ECU also increases, so it is difficult to make successful attacks successively.

Table 1 Restrictions according to the ECU state

ECU state	Access restriction for the CAN bus	Error flag
Error Active	none	6 dominant bits
Error Passive	transmission wait for 8-bit period during contiguous transmission	6 recessive bits
Bus-off	inhibit receiving and transmission	inhibit transmission

2) *Bus-off Attack Using a Stuff Error*

We introduce a bus-off attack using a stuff error, as shown in Fig. 3. By transmitting an error frame to the message transmitted by the target ECU, the TEC of the target ECU is increased, and it is transited to the bus-off state. Next, the attack procedure is shown. The attacker reads the CAN ID of the data frame flowing on the CAN bus and checks whether it is the one transmitted from the target ECU. If it is a frame transmitted by the target ECU, the attacker transmits an error frame. The target ECU detects a bit error according to the error frame transmitted from the attacker, and the TEC increases by 8. By repeating this process, TEC > 255 is set, and the target ECU is transited to the bus-off state.

3) *Bus-off Attack in One Frame*

We introduce a bus-off attack in one frame. As shown in Fig. 4, according to the CAN specifications, if the dominant state continues for a 14-bit period after error detection, the TEC is increased by 8. Furthermore, TEC is increased by 8 each time the dominant state lasts an 8-bit period. As a result, if the dominant state continues for a minimum of 255 bits (error bit 1 bit + 14 bits + 8 bits × 30), TEC > 255, and the target ECU is

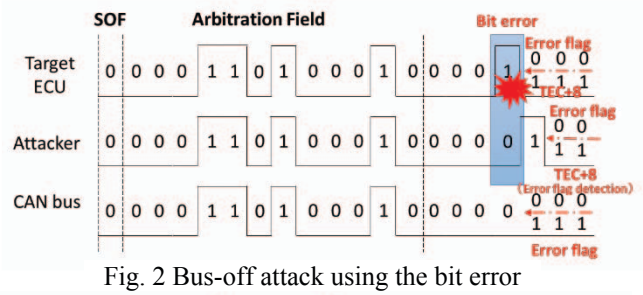


Fig. 2 Bus-off attack using the bit error

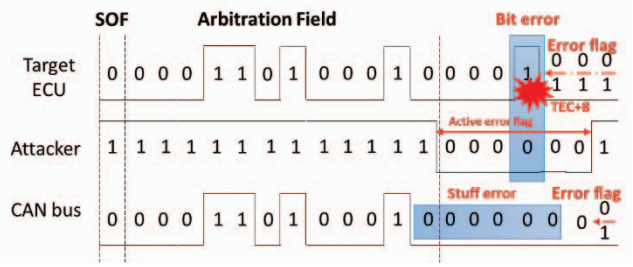


Fig. 3 Bus-off attack using the stuff error

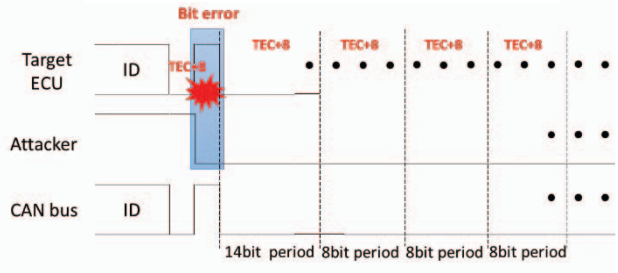


Fig. 4 Bus-off attack in one frame

transited to the bus-off state [6]. Because bit-stuffing is applied in the CAN, bits of the same level are not transmitted continuously for more than 6 bits. Thus, if the dominant state continues to be transmitted for 260 bits or more, the target ECU can definitely be transited to the bus-off state.

III. Proposed Attack Method

A. Theory of Attack

As shown in Fig. 5, when the target ECU that is detected by a specific CAN ID is transited to a bus-off state by using a bus-off attack, the authorized transmission ECU cannot detect the spoofing message. Now, the attacker transmits spoofing messages in the same cycle as the regular message. As a result, the receiving ECU cannot detect the spoofing. Since the receiving ECU receives only the spoofing message, it is impossible to detect anomalies from the receiving frequency.

B. Implementing the Proposed Method

Fig. 6 shows an example implementation of the proposed method. Here, the CAN bus level is sampled in real time by the sampling circuit with the start of frame (SOF) detection function, and the CAN ID of the data frame is detected. The detected CAN ID is checked to see if it was transmitted from the target ECU by the ID filter circuit. If it is the CAN ID of the message transmitted by the target ECU, the error frame transmission circuit transmits the error frame. In order to judge whether it is under attack at this time, the sampling circuit samples the data transmitted from the CAN controller for the latest 6 bits.

IV. Experiment and Discussion

A. Experimental Environment

The CAN bus experimental prototype consisted of an attack hardware and the target ECU. We used a field-programmable gate array (FPGA) board, microcomputer, CAN controller, and CAN transceiver to construct the attack hardware, shown in Fig. 6. A FPGA is used to analyze CAN frames and transmit error frames, and a microcontroller and a CAN controller are used to inject the spoofing message. Table 2 shows the experimental environment of the car. The CAN speed is set to 500 kbps, the target is the ECU that produces engine RPM CAN messages.

B. Result

As a conventional spoofing attack [4], spoofing messages were injected in the same cycle as the regular message without a bus-off attack. Fig. 7 displays the elapsed time from the previous message, the CAN ID, and the contents of the data part of the CAN message. The spoofing messages are enclosed in blue. The experimental results in Fig. 7(a) show that regular and spoofing messages were received alternately.

Next, a bus-off attack using a stuff error was performed on the target ECU. At the same time, a spoofing message was injected in the same cycle as the regular message. The experimental result in Fig. 7(b) shows that the spoof message was confused with the regular message. As shown by the red line, the reception interval after the regular message was received was less than 1 ms. The waveform in Fig. 8 confirms the successful transmission of the regular message. After a bit error occurred in the data frame transmitted by the attacker, an error frame including a passive error flag was transmitted.

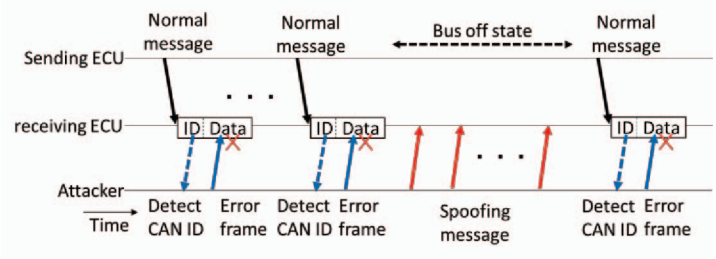


Fig. 5 spoofing attack using the bus-off attack

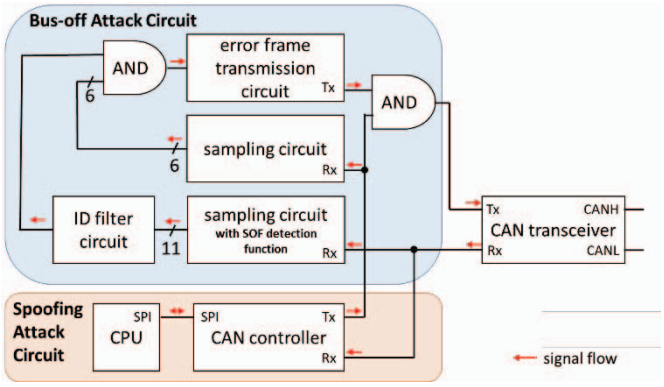


Fig. 6 Implementation of the proposed method

Finally, a bus-off attack in one frame was performed on the target ECU. At the same time, a spoofing message was injected in the same cycle as the regular message. The experimental result in Fig. 7(c) shows a spoofing attack with regular messages completely removed.

The results indicated that it was impossible to spoof the number of messages and cycle with the conventional spoofing attack and bus-off attack using the stuff error. However, the number of messages and cycle became complete when a bus-off attack in one frame was used.

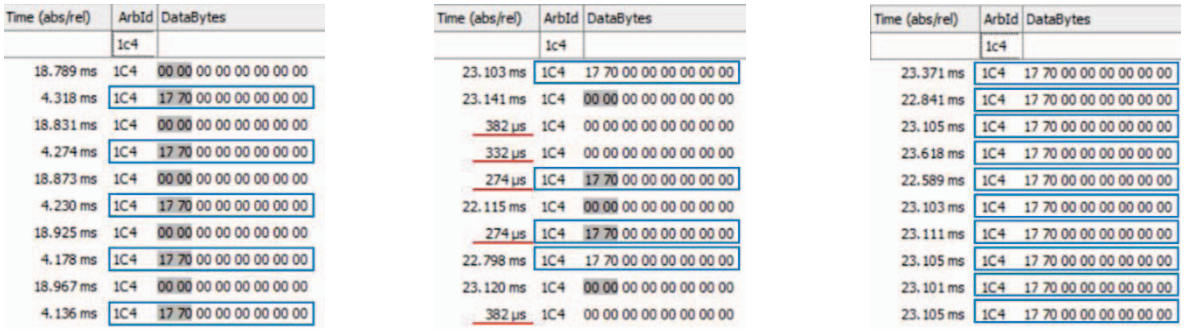
C. Experiment with a Car

We confirmed the effectiveness of the proposed method in the laboratory. Next, we evaluated it by using the CAN bus and ECU in a 2013-model Japanese hybrid vehicle. Table 2 presents the messages used. We performed a spoofing attack using a bus-off attack in one frame, which had the best result in the laboratory. The spoofing target was the engine speed, and the spoofing attack involved displaying an illegal value on the tachometer when the vehicle was stopped and idling. For a simple spoofing attack [4] with a conventional CAN injection, the spoofing message conflicts with the regular message representing 0 rpm. This causes the pointer of the tachometer to fluctuate between 0 and the spoofing value rather than point at the spoofing value the whole time.

When we attacked the car with the proposed method, the

Table 2 Experimental environment of the car

CAN Speed	500 kbps
Target ECU	CAN ID: 1C4 (engine RPM)
Regular message (period: 23 ms)	00 00 00 00 00 00 00 00 (RPM: 0)
Spoofing message (period: 23 ms)	17 70 00 00 00 00 00 00 (RPM: 6000)



Time (abs/rel)	ArbId	DataBytes
	1c4	
18.789 ms	1C4	00 00 00 00 00 00 00 00
4.318 ms	1C4	17 70 00 00 00 00 00 00
18.831 ms	1C4	00 00 00 00 00 00 00 00
4.274 ms	1C4	17 70 00 00 00 00 00 00
18.873 ms	1C4	00 00 00 00 00 00 00 00
4.230 ms	1C4	17 70 00 00 00 00 00 00
18.925 ms	1C4	00 00 00 00 00 00 00 00
4.178 ms	1C4	17 70 00 00 00 00 00 00
18.967 ms	1C4	00 00 00 00 00 00 00 00
4.136 ms	1C4	17 70 00 00 00 00 00 00

(a) Conventional spoofing attack

Time (abs/rel)	ArbId	DataBytes
	1c4	
23.103 ms	1C4	17 70 00 00 00 00 00 00
23.141 ms	1C4	00 00 00 00 00 00 00 00
382 µs	1C4	00 00 00 00 00 00 00 00
332 µs	1C4	00 00 00 00 00 00 00 00
274 µs	1C4	17 70 00 00 00 00 00 00
22.115 ms	1C4	00 00 00 00 00 00 00 00
274 µs	1C4	17 70 00 00 00 00 00 00
22.798 ms	1C4	17 70 00 00 00 00 00 00
23.120 ms	1C4	00 00 00 00 00 00 00 00
382 µs	1C4	00 00 00 00 00 00 00 00

(b) Spoofing attack using a stuff error

Time (abs/rel)	ArbId	DataBytes
	1c4	
23.371 ms	1C4	17 70 00 00 00 00 00 00
22.841 ms	1C4	17 70 00 00 00 00 00 00
23.105 ms	1C4	17 70 00 00 00 00 00 00
23.618 ms	1C4	17 70 00 00 00 00 00 00
22.589 ms	1C4	17 70 00 00 00 00 00 00
23.103 ms	1C4	17 70 00 00 00 00 00 00
23.111 ms	1C4	17 70 00 00 00 00 00 00
23.105 ms	1C4	17 70 00 00 00 00 00 00
23.101 ms	1C4	17 70 00 00 00 00 00 00
23.105 ms	1C4	17 70 00 00 00 00 00 00

(c) Spoofing attack using a bus-off attack in one frame

Fig. 7 Receiving periods of regular and spoofing messages

pointer of the tachometer became the impersonated value completely and never wandered, even though the engine was stopped. Because the regular message was not transmitted, only the spoofing message was transmitted. Therefore, the spoofing attack was effectively realized. There was no error indication in the car used for the experiment. Thus, the anomaly caused by the proposed method was not detected.

D. Discussion

In the spoofing attack using a bus-off attack with a stuff error, a regular message was also transmitted. As shown in Fig. 8, the attacker and target ECU transmitted at the same time. The attacker detected a bit error and transmitted a passive error flag. Since the attacker also transmitted when the ID was detected, the bus-off attack was not performed. The conditions for the successful transmission of the regular message are given below:

- The target ECU and attacker transmit at the same time
- $\lfloor TEC\ of\ Target/8 \rfloor \leq \lfloor TEC\ of\ Attacker/8 \rfloor$
- When the target ECU sends a recessive bit, the attacker sends dominant bits.

The expression in the second condition indicates that the attacker ECU first transited to the error passive state because of the bus-off attack using the bit error. This is the condition for successful transmission of regular messages. However, if the target ECU and attacker are exchanged, the transmission of the spoofing message will succeed. That is, while the target ECU is not in the bus-off state (i.e., the target ECU is in the error passive state), the spoofing message is transmitted. As a result, it is possible that an anomaly will be detected in the authorized transmission ECU. In the bus-off attack using the stuff error, because the target ECU retransmitted, the transmission timing had a high probability of overlapping with the attacker, which made it easier to satisfy the conditions. When the conditions were satisfied, the target ECU continuously transmitted the messages stored in the buffer. The reception interval of some messages was less than 1 ms.

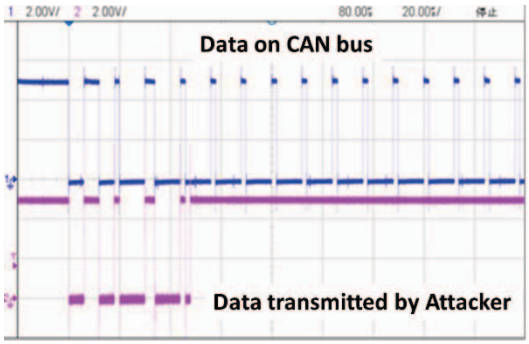


Fig. 8 Waveform when a regular message is successfully transmitted

In the spoofing attack using a bus-off attack in one frame, the target ECU was not allowed to retransmit. Therefore, the transmission of the regular message did not succeed because of the low probability of the transmission timings overlapping.

V. CONCLUSION

We proposed a spoofing attack method using a bus-off attack against an ECU, which is not detected by the authorized transmitting ECU and receiving ECU. We verified the attack in a simulated environment consisting of the attack hardware and ECU, and evaluated the effect of the spoofing attack on an actual car. The results indicated that the transmission of regular messages was completely prevented, and the percentage of spoofing messages could be made 100% with the proper cycle; that is, no error was detected in the ECUs of the car. We have verified the feasibility of the proposed method and the potential threat for an actual car. In the future, we will conduct studies to prevent vehicles from such attacks.

REFERENCES

[1] International Organization for Standardization, "Road vehicles, controller area network (CAN), Part 1: Data link layer and physical signaling," ISO IS11898-1, 2003.

[2] K. Koscher, et al., "Experimental Security Analysis of a Modern Automobile," Proc. 2010 IEEE Symposium on Security and Privacy, pp.447-462, May 2010.

[3] C. Miller, and C. Valasek, "Remote Exploitation of an Unaltered Passenger Vehicle," Black Hat USA 2015, pp.1-91, Aug. 2015.

[4] Takaya Ezaki, Tomohiro Date, and Hiroyuki Inoue, "An Analysis Platform for the Information Security of In-vehicle Networks Connected with the External Networks," Proc. The 10th International Workshop on Security, pp.301-315, Aug. 2015.

[5] Cho, Kyong-Tak and Shin, Kang G., "Error Handling of In-vehicle Networks Makes Them Vulnerable," Proc. 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS2016), pp.1044-1055, Oct. 2016.

[6] Ryota Kameoka, et al., "Bus-Off Attack against CAN ECU using Stuff Error injection," 2017 Symposium on Cryptography and Information Security (SCIS2017), pp.1-8, Jan. 2017. [in Japanese]

[7] A. Taylor, N. Japkowicz and S. Leblanc, "Frequency-based anomaly detection for the automotive CAN bus," 2015 World Congress on Industrial Control Systems Security (WCICSS), pp.45-49, 2015.