Electronic Preprint for Journal of Information Processing Vol.28

Regular Paper

Anomaly Detection Method "Cumulative Sum Detection" for In-Vehicle Networks

Jun Yajima[1,2,a)] Yasuhiko Abe[3] Takayuki Hasebe[1] Takao Okubo[2]

Received: March 26, 2019, Accepted: October 3, 2019

Abstract: This paper proposes cumulative sum detection, which can detect cyberattacks on Controller Area Network (CAN). Well-known existing attack detection techniques cause false positives and false negatives when there are long delays or early arrivals involving usual periodic message reception. The proposed technique can detect attacks with almost no false positives or false negatives, that is highly accurate even when there are a long delays or early arrivals. This paper evaluates the detection accuracy of existing techniques and the proposed technique using computer simulation with CAN data obtained from actual vehicles. By considering the evaluation result and the ease of parameter adjustment, we show that the cumulative sum detection is the best of these techniques.

Keywords: in-vehicle network, CAN, intrusion detection, reception cycle, cumulative sum

1. Introduction

Cyberattack countermeasures are very important for self-driving vehicles. In vehicle networks (IVN) based on Controller Area Network (CAN) protocol [1] is used in modern vehicles for vehicle control. However, the cyberattack countermeasures are not considered for CAN. Electronic Control Units (ECUs) connected to the IVN may operate in unintended ways when attack messages are injected into the IVN. Direct attack and indirect attack are the two categorized IVN attack methods.

In an indirect attack, the attacker hijacks a connection-ECU that connects to the external network by exploiting the vulnerability of the ECU and injects attack messages into the CAN from the external network through the ECU. We consider indirect attacks an extremely serious threat because the attacker need not to enter a car directly. A remote control indirect attack for actual vehicles was proposed in 2015 [2]. Because of this attack, the first recall was carried out for security reasons. In this paper, we focus mainly on indirect attacks.

Cycle detection [3], delayed-decision cycle detection [4], and Waszecki's method [5] were proposed to detect cyberattacks on CAN. These methods detect attacks using the reception cycle property that is common to many CAN messages, whereby they detect attacks by focusing on deviations in the reception cycle of CAN messages. However, in actual vehicles, periodic reception messages may be received that deviate from the designated periods. When long delays or early arrivals are caused, false positives are led in the attack detection. Additionally, other techniques are described in Refs. [6], [7].

1 Fujitsu Laboratories Ltd., Kawasaki, Kanagawa 211–8588, Japan
2 Institute of Information Security, Yokohama, Kanagawa 221–0835, Japan
3 Fujitsu Limited, Kawasaki, Kanagawa 211–8588, Japan
a) jyajima@fujitsu.com

Recently, we proposed a new detection method [8] called "cumulative sum detection". The cumulative sum detection can detect attacks with almost no false positives or false negatives for the perfect and almost periodic messages defined in Section 2. This means it is highly accurate, even when faced with long delays or early arrivals.

In this paper, we reintroduce the cumulative sum method. We describe the implementation method of the cumulative sum detection. We point out that a detector using our method should mention the reestablishment of counters for long span detection. We propose some reestablishment methods for effective detection. After that, we show the implementation algorithm using one of the reestablishment methods.

We describe other existing detection techniques and compare them with the detection accuracy of the cumulative sum. An evaluation based on computer simulation that uses an actual vehicle CAN data is applied. After that, the detection accuracy of each method is discussed. As a result, we show that the cumulative sum detection is the best of the above methods in the evaluation.

This paper is structured as follows. CAN is explained in Section 2 and assumed attacks are discussed in Section 3. The CAN traffic detection target used in this paper is described in Section 4 and existing methods are explained in Section 5. In Section 6, we propose the cumulative sum detection. The implementation method is shown in Section 7. Section 8 discusses the evaluation and its results using the actual vehicle's CAN data. Finally, the conclusion is described in Section 9.

2. CAN

2.1 Overview of CAN

CAN is a network protocol used in control systems, for example, the in-vehicle networks in cars. Network nodes called Electronic Control Units (ECUs) that use CAN communicate with each other by using the voltage difference between two commu-

nication lines. CAN is a broadcast communication protocol, so data transmitted from any ECU reaches all ECUs connected with the same bus. Each bit of a CAN message is either dominant (0) or recessive (1). The dominant part of the message is given priority when messages are transmitted to communication lines by two or more ECUs at the same time. CAN has four data format types (data frame, remote frame, error frame, and overload frame). This paper focuses on the data frame, so the details are explained as follows.

2.2 Data Frame

There are an 11-bit ID field in the standard format and a 29-bit ID field in the extended format. Although this paper only mentions the standard format, the discussion is also applicable to the extended format. We call this ID field value "CAN-ID". CAN-ID shows the meaning of the message and is used for communication arbitration. Communication arbitration is a mechanism whereby only the high priority message is processed when two or more ECUs transmit data frames at the same time. Because the dominant is given priority over the recessive, the message where the dominant appears first in the most significant bit (MSB) of CAN-ID is processed first in all transmitted frames. When the dominant appears in the MSB of CAN-ID simultaneously, priority is similarly determined by the subsequent bit. The transmission node stops transmitting the transmission message if the CAN-ID of the message which is sent from another ECU at the same timing has higher priority than the transmitting message. This mechanism is called message arbitration. In general, CAN-IDs are operated such that the message with the same CAN-ID is not used when transmitted ECUs are different. In addition, the messages that are received by each ECU are predetermined based on the CAN-ID. Messages with different CAN-IDs are ignored by each ECU.

2.3 CAN Message Periodicity

On the CAN data frame, many messages are transmitted at the designated transmission cycle determined for each CAN-ID. There are various cycle lengths, which range from about ten milliseconds to ten seconds. Additionally, a few messages are transmitted non-periodically. We define CAN messages as the following three types.

- Periodic messages:
  - Perfect periodic messages: Message reception intervals may be temporary slightly deviated. Namely, message reception intervals are recovered to normal status in a fraction cycle. The reception number of messages for each CAN-ID is constant for short time observation. There is no shortage or excess of messages.
  - Almost periodic messages: Message reception intervals may temporarily have large deviations. One example of a cause of temporary deviation is message arbitration. Namely, message reception intervals is recovered to normal situation in some cycles. The reception number of messages for each ID is constant for long time observation. There is no shortage or excess of messages.
- Event based periodic messages: Messages are sent to the CAN-bus periodically in normal situations. However, when some events occur in the ECU, event-driven messages are sent to CAN-bus. From then onwards, messages are sent periodically again to CAN-bus.
- Non-periodic messages: Message reception intervals may be large deviated. Namely, message reception intervals cannot be decided. The reception number of messages for each CAN-ID is not constant for long time observation.

We focus on the perfect and almost periodic messages that include long delayed and early arrived messages. In general, these messages account for more than at least half of the all messages. We assumed that attacks against the event-based and the non-periodic messages are detected by other techniques. These are therefore outside of the scope of this paper.

3. Attack Assumption

Cyberattacks against IVNs are classified as one of the following two varieties. In this paper, we focus on detecting the indirect attack, although the proposed method can detects some types of direct attack.

3.1 Direct Attack

This attack uses an attack device that connects to the IVN. The attacker connect the attack device directly to the CAN entry point, such as the OBD-II port. Modified hardware from a general CAN controller may be used as this attack device. In this case, attack detection using the electrical signal level may be needed. If the attacker can enter the car and connect the attack device to the IVN's entry point, this attack is a big threat because the attacker can send arbitrary messages to the in-vehicle network. However, entering the car without the key is generally difficult. Taking countermeasures to prevent intrusion into the car (locking doors, etc.) makes such attacks difficult.

3.2 Indirect Attack

This attack takes advantage of the vulnerabilities of the communication equipment that connects to the Internet, etc. The attacker hijacks the connection-ECU that connects to the external network by exploiting vulnerability of the ECU, and injects attack messages to the IVN from the external network through the ECU. Although such attacks are limited compared with direct attacks because the ECU cannot be modified at the hardware level, the attacker need not enter the vehicle and can attack it remotely. To prevent such attacks, it is important to develop ECUs without vulnerabilities. As with personal computers or smartphones, latent vulnerabilities can be discovered several years after the manufacturer launches the product. Because it is difficult to eliminate all vulnerabilities in ECU, this attack is a big threat. This attack comes from the external network, so the attack messages are transmitted from the general CAN controller installed in the hijacked ECU. In terms of attack detection, the attack can be detected based on the messages received by the CAN controller.

4. CAN Traffic to be Detection Target

As explained in Section 2, messages with a high priority CAN-ID flow to CAN when two or more messages are transmitted from different ECUs at the same time. When a certain ECU transmits a

message M_1 whose CAN-ID has the higher priority, if another ECU has already started transmitting the ID field of message M_2 whose CAN-ID has the loweer priority, the message M_1 is transmitted after message M_2 has been transmitted. Therefore, there is a delay in message transmission from the ECU to the IVN. The methodology that forecasts the maximum collision delay time in the message transmission time is known and is reported in Refs. [9], [10], [11], etc. In the existing detection methods explained in Section 5, their targets are the perfect periodic messages. By using the existing methods for small delayed and early arrivaled messages, false negatives can be minimized and attacks detected with high accuracy. However, while there are periodically transmitted messages, in actual vehicles, long delays and early arrivals may sometimes occur. In these cases, many of these methods cannot detect attacks correctly. In this paper, we aim to construct a method that can detect attacks with extremely high accuracy with very few false negatives, even in the situations with large delays and early arrivals. Namely, our targets are the perfect periodic messages and the almost periodic messages.

5. Related Methods

This section explains three well-known detection methods for periodic messages on CANs. As explained in Section 4, most of these may have false positives or negatives in the situation of significantly delayed arrivals and early arrivals.

5.1 Cycle Detection

This method observes only the interval in reception times between two consecutive messages. We call this method cycle detection. For the cycle detection method, Ref. [3] is known. The cycle detection investigates whether the interval of messages differs from the normal communication. Specifically, the arrival time of the following message under ideal reception conditions is ascertained and the permissible boundary is set based on this. The subsequent message is judged as a normal message if its reception time is within the permissible boundary, and detected as an attack if it falls outside of the boundary. The cycle detection is shown in **Fig. 1**. In this example, when the interval of the reception times of two consecutive messages exceeds the predetermined permissible boundary, this method detects this situation as an attack. The permissible boundary is updated by using the current reception time of the message when the time is within the permissible boundary.

5.2 Delayed-decision Cycle Detection

The delayed-decision cycle detection is a detection algorithm proposed by Otsuka et al. [4]. This method observes the interval of reception times for about 3 consecutive messages. This method is an algorithm that considers the possibility of fluctuation of the message transmission cycle. This method is explained in **Fig. 2**. In this algorithm, two parameters named α and β are used. α is a parameter used to decide which message is to be a detection target. The reception time t_1 of a certain message is not detected as an attack when the time interval between the previous reception time t_0 and t_1 exceeds $T-\alpha$ (T is the cycle of the message). When t_1 is less than $T-\alpha$, the detection judgement is reserved and reception of the next message is awaited. When the reception time of the next message is less than $T+\beta$, this situation is detected as an attack. As a result, this method can detect attacks that occur during delays and early arrivals.

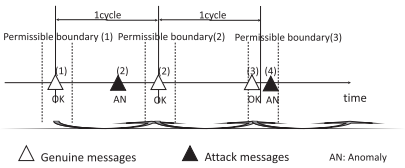


Fig. 1 An example of the cycle detection.

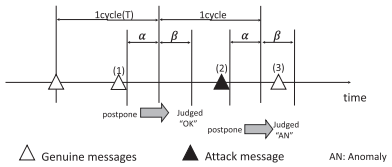


Fig. 2 The delayed-decision cycle detection.

5.3 Waszecki's Method

This method is proposed by Waszecki et al. [5]. In this method, a detector derives worst case jitter j which means maximum deviation from reception cycle. After that, it derives the minimum time interval of consecutive messages δ, burst capacity ν, and decrement value of counter ρ. If the detector can derive it, this method can detect attacks with high accuracy. In the detection algorithm, the detector calculates the counter and timer. The counter is subtracted ρ when message is received. The timer indicates the timing of increments of the counter. If the counter is less than zero, the situation is judged as an attack. This method seems similar to our method at first sight. However, this method differs from our method in some features. The differences are as follows.

- Need of the calculation of jitter: In their method, all parameters must be set correctly. Much of them can be calculated using the derivation formula shown in the paper. Only jitter must be determined by the detector. Similarly, in our method, all parameters must be set correctly. However, in most cases, all parameters can be selected easily in our method.
- Treatment of counters after detection: Treatment of counters after attack detection seems to be not described. After the attack detection, the values are sure to have less of a decrease than during normal use. Therefore, it may cause the false positives or negatives when continuing the processing their method without increasing the value of the counter.

6. Proposed Method (Cumulative Sum Detection)

The proposed method is explained in this section. We call this method cumulative sum detection (CSD). The CSD can detect attacks when messages are the perfect or the almost periodic messages described in Section 2.3.

6.1 Basic Idea

The CSD uses two counters. The first is counter n, which in-

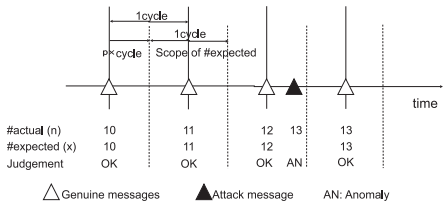


Fig. 3 Basic cumulative sum detection concept.

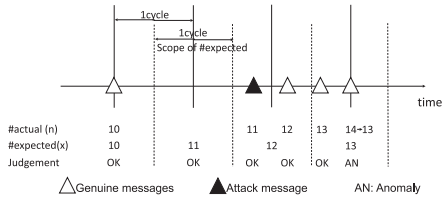


Fig. 4 Behavior of the cumulative sum detection in a delay situation (an attack is injected).

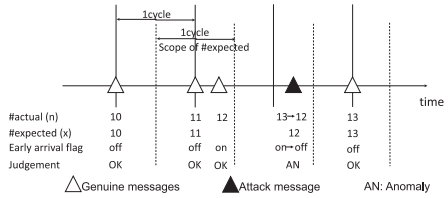


Fig. 5 Behavior of the cumulative sum detection in an early arrival situation (an attack is injected).

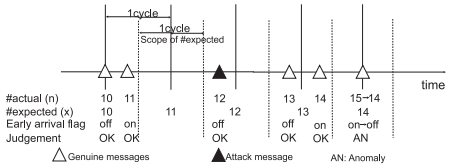


Fig. 6 Behavior of the cumulative sum detection in delayed and early arrival situations (an attack is injected).

dicates the number of actual reception messages. This counter is incremented when a message is received. The second counter is x. x indicates the number of expected reception messages. This counter is incremented at each passage of a cycle. As a basic idea, this method detects a situation where $n > x$ as an attack. This method utilizes the reception time t of the first message, the cycle T, and the position ratio p of counting boundary of x. The increment timing of x is defined as $t + p \times T + k \times T$ ($k = 1, 2, ...$). For p, we can use 0.5 empirically. An example of attack detection when an attack is injected into a periodic transmission message is shown in **Fig. 3**. In this example, the attack is injected when the number of the expected reception message is 12. The relationship of the two counters becomes $n > x$ and the attack is detected. After the attack is detected, the attack detection can be continued by decrementing the counter n.

6.2 Delayed Messages

The basic idea shown in Section 6.1 can be used when genuine messages are delayed. **Figure 4** shows an example of the CSD where genuine messages are delayed. In this example, two messages delay when the numbers of the expected reception messages are respectively 11 and 12. The delay is recovered when the number of the expected reception messages is 13. In this example, the attack is injected immediately before the recovery of the delayed message. The number of actual reception messages n exceeds x when x is 13 and the situation is detected as an attack. On the other hand, when the attack is not injected in the same situation, there are no false positives because n is smaller than x until the delay is recovered.

Using the proposed method, reliable detection is possible even when the detected time of attack is not exactly the same as the attack time. This characteristic is the same as many existing methods. Therefore, we allow this characteristic.

6.3 Early Arrival of Messages

The basic idea shown in Section 6.1 cannot be used when genuine messages arrive earlier than scheduled cycles. In the proposed method, an early arrival flag is used to detect an attack accurately during an early arrival. When $n = x + 1$ due to the early arrival of a message, the flag becomes ON, and is judged to be normal temporarily. After that, when $n = x + 1$ and this flag is ON when the messages are received, the situation is detected as an attack. When the attack is not injected during an early arrival, $n \leq x$ is satisfied. The early arrival flag is set to OFF. A detailed algorithm is described in Section 6.5.

Figure 5 shows an example of the CSD for an early arrival. In this example, an early arrival occurs when the number of expected reception messages is 11, $n = x + 1$ is satisfied, and the early arrival flag is set to ON. Thereafter, $n = x+1$ is also satisfied when the number of expected reception messages is 12, the early arrival flag is ON, and the situation is detected as an anomaly. On the other hand, in the same situation without an attack, although the early arrival flag is set to ON when $n = 12$, the flag is set to OFF when $n = 13$. The detection judges the message genuine. This judgment causes no false positives.

In order to allow two or more early arrivals, the early arrival start timing must be stored with an early arrival flag. If number of early arrivals e are allowed, the early arrivals are recovered to normal situation until e cycles after the start timing. When the following three conditions are all met when a message is received, the situation is detected as an anomaly.

(1) $x < n \leq x + e$ is satisfied.
(2) The early arrival flag is 'ON'.
(3) $n >$ 'the early arrival starting location' $+e$ is satisfied.

6.4 Combinations of Delayed and Early Arrival Messages

Even if delays and early arrivals are combined, we confirmed that there are almost no false positives or negatives when the idea explained in Section 6.3 is used. **Figure 6** shows an example of the behavior of the CSD when an attack is injected with delayed and early arrival messages. In this example, the attack is injected when the number of expected reception messages is 12. The attack is detected when the number of the expected reception messages is 14.

6.5 Proposed Method

The proposed method can be used for only messages whose CAN-IDs are same. Therefore, some CAN-IDs of the monitoring target are decided before using the proposed method. The method

is applied to the monitoring targets independently whose CAN-IDs are different. In the CSD, the following expression is evaluated at each reception timing of i-th message t_i ($i = 0, 1, 2, ...$).

$$N_i - X_i > e \tag{1}$$

$$N_i - X_i > 0, N_i - X_i \leq e \tag{2}$$

where $N_0 = 0$, $X_i = \lfloor \frac{t_i - t_0 + p \times T}{T} \rfloor$.

- X_i number of expected reception messages
- N_i number of actual messages
- e allowable early arrival

When a message is received, $N_i = N_{i-1} + 1$. After that, the situation is judged as 'Normal' or 'Abnormal (Anomaly)' depending which expression is satisfied.

1. if the expression (1) is satisfied, the situation is judged as 'Abnormal'.
2. if the expression (2) is satisfied, the situation is judged as 'Normal' temporarily. After that, if it continues for e cycles after the expression (2) is satisfied, the judgement is changed to 'Abnormal'.
3. Otherwise, the situation is judged as 'Normal'.

If i-th message is judged as 'Abnormal', $N_i = N_i - 1$,

6.6 Disscussion of the Detection Capabilities of the Cumulative Sum Detection and the Existing Methods

On all the related methods and proposed method, the timing for anomaly detection are not exactly same as the reception timings of attack messages. Therefore, the number of false positives and negatives derived from timing inconsistency of them are not useful for accuracy comparison. We simply define false positive the exceeding number of detections against attack messages in this paper.

6.6.1 Cycle Detection

In this method, genuine messages are incorrectly detected as attack messages when the reception cycle of genuine messages is significantly deviated by large delays and early arrivals, namely the deviation exceeds the permissible boundary. Moreover, this method may overlook the attack when the permissible boundary is too wide. Therefore, parameter adjustment of the wideness of the permissible boundary is very important. This method cannot detect attacks correctly in a reception deviation situation.

6.6.2 Delayed-decision Cycle Detection

In this method, when an attack message is received just behind a genuine message M_{g1} and reception of the next genuine message M_{g2} is delayed, a false negative is caused. More specifically, the delay means that time interval of M_{g1} and M_{g2} exceeds $T + \beta$. Moreover, the method of deciding upon α and β is not shown. These parameters should be tuned by hand for each CAN-ID. Generally, this work is time consuming. This method can detect attacks correctly in the situation of small and medium reception deviation. The detection timing may be delayed in some situations.

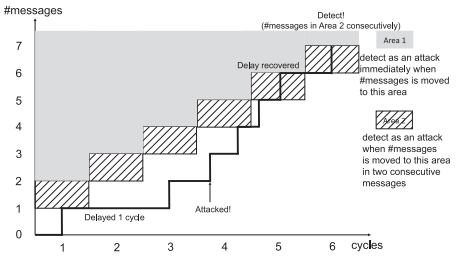


Fig. 7 Detection example of the cumulative sum detection in delayed situation.

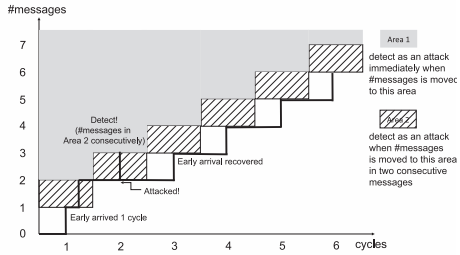


Fig. 8 Detection example of the cumulative sum detection in early arrived situation.

6.6.3 Waszecki's Method

In this method, the value of worst case jitter is very important for accurate detection. If the worst case jitter value j is not set correctly, false positives or negatives may be caused. Moreover, when the worst case jitter is more than 1 cycle, we cannot derive the correct values of other parameters especially δ. Because δ is derived by the greatest common divisor (GCD) of the cycle T and $T - j$. When $T < j$, the value must be derived by the GCD of T and negative value $T - j$. Incorrect parameters may be derived. Therefore, we think that their method cannot detect attacks accurately in large jitter situations. This method can detect attacks correctly in the situation of small, medium, and large (< 1 cycle) reception deviation, when parameters are adjusted correctly. The parameter adjustment is not easy in some cases. The detection timing is late in delayed situations.

6.6.4 Cumulative Sum Detection

The detection example of the CSD is shown in **Fig. 7** and **Fig. 8**. In these figures, if the number of cumulative sum of actual message reception (NCA) falls under Area 1, the CSD detects the situation as an attack immediately. If the NCA falls under Area 2 for two consecutive messages, the CSD detects the situation as an attack. After the detection, the NCA is decremented by 1. On the other hand, in Waszecki's method, if the NCA falls under Area 1 or Area 2 at one time, their method detects the situation as an attack immediately. However, they didn't show the decrementation method of the NCA after the detection. We assumed that there is no shortage of messages. Therefore, when large delay and/or early arrival is recovered, the NCA is recovered to normal state. When an attack is injected during delayed and early arrived situation, the attack will be detected immediately after recovering from the delayed and early arrived situation to normal. In our method, allowable early arrival e is important. This parameter signifies the maximum number of early arrival messages with in e cycles. When the number of early arrival or the number of recovering cycles exceed e, a false positive is caused. How-

Table 1 Comparison of the existing methods and the proposed method.

	Cycle [3]	Delayed-decision Cycle [4]	Waszecki [5]	Cumulative Sum (proposal)
Small deviation on cycle *1 (< 16.6%)	Good	Good	Good	Good
Medium deviation on cycle *1 (< 50%)	Bad	Good	Good	Good
Large deviation on cycle *1 (> 50%)	Bad	Not Good *3	Fair *4	Good
Parameter adjustment	Difficult	Difficult	Difficult	Easy
Real time characteristic *2	Good	Fair *5	Fair *5	Fair *5

ever, we can set enough e empirically because there is very little possibility of a great increase in early arrivals. Actualy, in many time evaluations, we set $e = 1$, and it cause no false positives. P means the boundary prameter of counting. In our method, the number of expected reception messages is increased by 1 every a cycle without depending on P. Therefore, we can choose P empirically. This method can detect attacks correctly in the situation of small, medium, and large reception deviation, when parameter e is set correctly. We think that when the boundary of counting is set to half of the reception cycle, the detection accuracy is highest for periodic messages. The parameter adjustment is easy in all cases. The detection timing may be delayed in delayed situations.

This discussion is summarized in **Table 1**. Based on this table, the CSD is the best detection algorithm among four methods.

7. Implementation

7.1 Reestablishment of the Counters and the First Timestamp

In order to use the proposed algorithm effectively, it is preferable to reestablish the counters x and n. Because there may have small gap between the installed cycle information for the detection algorithm and the actual cycle of the vehicle, false positives or negatives may be caused for long span using of the proposed method. In order to solve this problem, the reestablishment processing is needed. The processing is $x = x - n$, $n = 0$, and adjusting the counting boundary. The execution condition of this processing have some variations. Some of them are described as follows.

(1) Over predetermined number of receptions
The reestablishment processing is executed when the number of message receptions exceeds predetermined threshold r. This is very simple condition, however, false positives or negatives may be caused when attacker knows the threshold value and focuses attack messages on the reestablishment timings.

(2) Over random number of receeptions
The reestablishment processing is executed when the number of message receptions exceeds randomly determined threshold r_j. This method decreases the success probability of the attack described in the above.

(3) Limited reception interval
The number of message receptions exceeds threshold that is predetermined or randomly determined. In addition to it, the reestablishment processing is executed when the interval of reception timing of the last message and the one before message is within another threshold q. This method dramatically decreases the success probability of the attack described in the above.

*1 This deviation is independent from the message priority (CAN-ID).
*2 This is the shortness of the detection delay from message reception.
*3 This means the DDC method seems not to be good property for the situation of large deviation. But detection ability seems to be better than the cycle method.
*4 This means the detection may cause false negative in case of the worst case jitter is over 1 cycle.
*5 This means the detection delay is only few cycles. However, we think this delay is not a major problem.

7.2 Implementation Algorithm with Reestablishment

Parameters in the proposed algorithm are as follows. In this algorithm, the allowable early arrival e is assumed to 1 and the allowed recovery cycle of the early arrival w is also assumed to 1. It is easy to enhance the algorithm to w ($\geq e$) at $e \geq 2$. In this algorithm, user should pre-determine the reestablishment parameter r, reestablishment condition q, and increment timing parameter p. For example, these values are casually determined like $r = 10$, $q = 0.1$, and $p = 0.5$. The algorithm consists of pre-computation, main function and reestablishment processing. The pre-computation is executed one time at first reception of message for each CAN-ID. The main algorithm is executed every reception timing with $i \geq 1$ for each message with same CAN-ID.

Parameters

i	order of message reception ($i = 0, 1, 2, ...$)
r	number of reception messages before reestablishment. (e.g., r=10) (pre-determined)
p	parameter of the increment timing whereby $0 < p < 1$ (generally, $p = 0.5$ is preferable). (pre-determined)
T	reception cycle of message (generally, this is average cycle) (pre-determined)
t_i	reception timestamp of i-th message whose CAN-ID is ID
d	next expected reception time
n	actual number of message reception
x	expected number of message reception
f	early arrival flag
ID	CAN-ID
J	Judgement result for each reception

Pre-computation
Input: p, T, t_0
Output: d, n, x, f
$d = t_0 + (1 + p) \times T$
$n = 0, x = 1, f = off$
Output(d, n, x, f)

Main function
Input: $r_i, p, T, t_i (i \geq 1), d, n, x, f$
Output: J, d, n, x, f
$n = n + 1$
while $t_i > d$ do
  $x = x + 1, d = d + T$
end while
if $n \leq x$ then
  $J \Leftarrow$ 'Normal'
  $f = off$
else if $n = x + 1$ then
  if $f = off$ then
   $J \Leftarrow$ 'Normal'
   $f = on$
  else
   $J \Leftarrow$ 'Abnormal'
   $n = n - 1$
   $f = off$
  end if
else
  $J \Leftarrow$ 'Abnormal'
  $n = n - 1$
  $f = on$
end if
Call the following "Reestablishment processing"
Output(J, d, n, x, f)
Reestablishment processing
Input: $d, n, x, p, T, r_i, t_i, t_{i-1}, q$
Output: d, n, x
if $n \geq r_i$
  if $t_i - t_{i-1} < q$
   $x = x - n, n = 0$
   $d = t_i + p \times T$
  end if
end if

8. Evaluation

8.1 Overview

To confirm the comparison result shown in Table 1, we evaluated the detection accuracy of each method using PC simulation. In this evaluation, the sending and receiving of CAN communication was simulated on a PC. We obtained genuine data from an actual vehicle, and many attack messages were injected into them. After the evaluation, we considered and compared the accuracy of the attack detection methods.

8.2 Data Preparation

We drove a car while taking the CAN-log from the OBD-II port over ten minutes. After that, we looked for the perfect periodic messages and the almost periodic messages from the log. Finally, we found some variations of such genuine data, and we decided to use two (Data1 and Data2 in **Table 2**) of them. Each data has mutually different CAN-ID. In the evaluation, initially, the data (no attack is injected) is evaluated to confirm whether a false positive occurs. Next, attack messages are injected into Data1 and Data2, and each attack-injected Data1 and Data2 is evaluated to confirm whether the false positives and negatives occur. In the data preparation task, we made many attack-data based on three variations shown in **Table 3**. The first type of the attack is single attack. An attack message is injected at random timing into one genuine data. We prepared 10000 data of this data type for each genuine data. In the second type of the attack, 100 attack messages are injected to one genuine data. Each attack message is injected at random timing. We prepared 10000 data of this data type for each genuine data. In the third type of the attack, 100 periodic attack messages are injected to one genuine data. The injecting timing of the first attack message is determined at random. We prepared 10000 data of this data type for each genuine data.

Table 2 Types of genuine data.

Name	Length	Cycle	Features
Data1	600 sec	9,999 usec	Few delayed arrival and early arrival messages.
Data2	600 sec	9,982 usec	Many delayed arrival and early arrival messages.

Table 3 Patterns of attack data.

Name of Pattern	Number of data set	Features of attack message(s)
Pattern1	10000	1 message with random injection times in 1 data set
Pattern2	10000	100 messages with random injection times respectively in 1 data set
Pattern3	10000	100 messages at the appropriate cycle in 1 data set. The 1st message is injected at random times.

8.3 Parameter Adjustment

A number of variations exist in the cycle detection. In this evaluation, the method [3] explained in Section 5.1 is adopted. Parameter adjustment is needed for three existing methods.

8.3.1 Cycle Detection

One example is described in Ref. [3] as ±1 ms about the permissible boundary of the cycle detection. Using this parameter, evaluation of Data1 (no attack is injected) found a large number of false positives (60,000 messages or more across 600 seconds). We then modified the specification of the cycle detection. In the new specification, the permissible boundary is always updated when a message is received, even if the reception time is not within the permissible boundary. As a result, the number of false positives decreased considerably. We started the parameter of permissible boundary with a very low value. And we repeated experiments using higher slightly value until false positives are disappeared. Based on many times experiments using this method, We adjusted the parameter of permissible boundary to ±8% and ±42%, for Data1 and Data2, respectively.

8.3.2 Delayed-decision Cycle Detection

Recommended parameters have not been described in Ref. [4] for the delayed-decision cycle detection. Therefore deciding the parameter value is very difficult. This detection method consists of two phases, namely the checking phase using α and the checking phase using β. At first, we thought that the detection capability seemed to be best when α was 0, because when $\alpha = 0$,

Table 4 Evaluation result.

Data	Attack pattern in Table 3	Number of total genuine messages	Number of total attack messages (NA)	Number of detected messages as anomaly (ND)					
				Cycle	Delayed-decision cycle 1	Delayed-decision cycle 2 (optimized)	Waszecki's method 1	Waszecki's method 2 (optimized)	Cumulative Sum (proposal)
Data1	Pat.1	676000000	10000	18406	13612	10000	82394523	10000	10000
	Pat.2	676000000	1000000	1839102	1360543	999972	83390855	1000000	1000000
	Pat.3	676000000	1000000	1842671	1003663	1000000	83267634	1000000	1000000
Data2	Pat.1	706700000	10000	11641	10901	9988	10000	10000	10000
	Pat.2	706700000	1000000	1160431	1085404	997810	1000000	1000000	1000000
	Pat.3	706700000	1000000	1161273	1000802	998348	1000000	1000000	1000000

By the definition in Section 6.6, when $N_{DA} = ND - NA > 0$, N_{DA} shows the number of false positives. When $N_{DA} < 0$, $-N_{DA}$ shows the number of false negatives.

the checking phase using α will be skipped. However, after we set these parameters, many false positives occurred in the experiments. The result is shown as "Delayed-decision cycle detection 1" in **Table 4**. So, we adjusted the parameters by many trial experiments. Finally, we decided to adjust α to 8% of the reception cycle for Data1 and 42% of the reception cycle for Data2. After that, we decided to adjust $\beta = 92\%$ and 61% for Data1 and Data2 respectively. The result is shown as "Delayed-decision cycle detection 2" in Table 4.

8.3.3 Waszecki's Method

The derivation method of the worst case jitter is not described in Ref. [5] in detail. We think that this value can be derived by the formula (3) (T is reception cycle). This formula means that the worst case jitter is defined as the maximum difference between the message interval and message reception cycle.

$$j = \max_i(|T - (t_i - t_{i-1})|) \tag{3}$$

By using the above formula, we derived $j = 72$ (usec) for Data1 and $j = 4{,}144$ (usec) for Data2. However, many false positives occurred in the evaluation for Data1. The result is shown as "Waszecki's method 1" in Table 4. So, we think the following is another derivation formula of j.

$$j = \max_i(|t_i - (t_0 + i \times T)|) \tag{4}$$

The formula (4) means that the worst case jitter is defined as the maximum difference between the reception time and the expected normal reception time which is derived from first reception time. By using the formula (4), we derived $j = 5{,}062$ (usec) for Data1 and $j = 5{,}154$ (usec) for Data2. After that, this method works well. The result is shown as "Waszecki's method 2" in Table 4.

8.3.4 Cumulative Sum Detection

Using the CSD on both Data1 and Data2, no false positives occurred when the allowable early arrival e and the allowed recovery cycle of early arrival w were set to 1 and 1 respectively. P was set to 0.5 empirically. So, we adopted these values. On the reestablishment method, we adopted (1) in Section 7.1. We used 20 for the threshold r.

8.4 Evaluation Results

The results of this evaluation are shown in Table 4. In the cycle detection, up to two detections occur for one attack injection. Therefore, we cannot judge whether the cycle detection is correct because we cannot know how many detections occur for the attacks. In the delayed-decision cycle detection, by the difficult parameter adjustment, the results become better than the results using the simple parameter decision. However, this result is not optimal yet. In Waszecki's method with very optimized parameter and the CSD, the number of detections are equal to the number of attack injections for all data sets. Therefore, these method can detect attacks with high accuracy.

Reception time of Data3

11.000000
12.000000
13.000000
14.000000
15.000000
15.800000 (early arrived from 16.0)
15.900000 (large early arrived from 17.0)
17.000000 (attack)
17.100000 (attack)
17.200000 (attack)
17.300000 (attack)
17.400000 (attack)
18.000000
19.000000
20.000000

Reception time of Data4

11.000000
12.000000
13.000000
14.000000
15.000000
15.800000 (small deviation)
15.900000 (1 early arrived)
16.100000 (2 early arrived)
16.200000 (attack)
19.000000
20.000000

Fig. 9 Data for appended evaluation.

8.5 Consideration

8.5.1 Comparison of the Cumulative Sum Detection and Waszecki's Method

In this evaluation, the detection accuracies of the CSD and well optimized Waszecki's method were the same. By the discussion at Section 6.6, these methods may cause a false positive or negative in the specific situation. In order to confirm this, we executed two small additional evaluations. In this evaluation we made two data whose reception cycle is 1 second. Data for additional evaluations are shown in **Fig. 9**. In the first additional evaulation, we made Data 3 which includes large early arrived message whose jitter exceeds one cycle in Waszecki's method. As a result, the CSD caused no false positives or negatives. Also, Waszecki's method caused no false positives. However, their method caused 5 false negatives. In the second additional evaluation, we made Data 4 which includes 2 early arrived messages that exceed the allowable early arrival e in the CSD. As a result, the CSD caused 1 false positive and no false negatives. On the other hand, Waszecki's method caused no false positives and 1 false negative. The reason is that this situation also causes large jitter for Waszecki's method. Therefore, we think the CSD is a little better in detection accuracy.

8.5.2 Difficulty of Parameter Adjustment

From the results shown in Section 8.4. parameter adjustment seems to be difficult for the cycle detection, the delayed-decision cycle detection and Waszecki's method. These methods need parameter adjustment by investigating the maximum deviation. The maximum deviation is derived by collision delay in many

cases. However, in some cases, maximum early arrival and delay of messages are caused by the message deviation of itself. The collision delay can be predicted by real-time scheduling method like [10], [11]. On the other hand, it is difficult to predict the maximum early arrival and delay of messages itself. As one solution, long communication log can be used for the prediction. In general, it is thought that using the long-term log in this preliminary investigation makes for highly accurate detection. However, there is no guarantee that the maximum early arrival or delay of the cycle reception occurs in the preliminary investigation log. When the prediction is wrong, false positives and negatives may be caused. In the evaluation using Waszecki's method, we derive the worst case jitter from the log file. Therefore, the jitter was specialized for these evalations. This cause very highly accurate detection in these evaluations. In general, the derivation of the worst case jitter is difficult. On the other hand, the parameter adjustment of the cumulative sum detection is very easy. Very highly accurate detection is achieved without considering sencitive parameter adjustment.

Consideration of the deviation from the cycle reception is not needed for the CSD; only the number of early arrivals must be considered. This seems to be much easier than adjusting parameters in the existing methods. Overall, the CSD has the best properties among the discussed methods. In this evaluation, as explained in Section 6.2, we admit that the judging time is not exactly the same as the attack injection time, because all the methods described in this paper detect attacks at times that may not be exactly the same as the attack injection time.

The CSD and the existing methods belong to the technique for observing message cycle. When an attacker cancels normal messages and injects attack messages in same timings, the attack cannot be detected by these techniques. These techniques also cannot detect message eavesdropping. Those attacks are not within in the scope of this paper.

9. Conclusion

This paper proposes cumulative sum detection (CSD) that detects attacks with very few false positives or negatives when large delays and early arrivals occur and the reception cycle of the periodic transmission message is biased. We point out the detector should mention the reestablishment of the counters. We propose the implementation method using the reestablishment method. We evaluate and compare the detection accuracy of the cumulative sum detection, the cycle detection, the delayed-decision cycle detection, and Waszecki's method. The cumulative sum detection and Waszecki's method had the best detection accuracy among the four methods. However, Waszecki's mwthod needs precisely parameter adjustment for high accuracy detection. On the other hand, the parameter adjustment for the cumulative sum detection is almost unnecessary. We conclude that the cumulative sum detection is an excellent method in the viewpoint of the detection accuracy and the parameter adjustment.

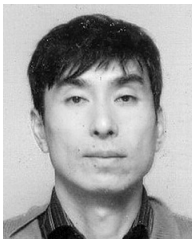
References

[1] ISO11898: Road vehicles – Controller area network (CAN) (2003).
[2] Miller, C. and Valasek, C.: Remote Exploitation of an Unaltered Passenger Vehicle, *BLACKHAT2015* (2015).
[3] Kishikawa, T., Matsushima H., Haga, T., Maeda, M., Umigami, Y. and Ujiie, Y.: In-Vehicle Network System, Electronic Control Unit, and Irregularity Detection Method, Publication Number WO/2015/170451, International Applications (2015).
[4] Otsuka, S., Ishigooka, T., Oishi, Y. and Sasazawa, K.: CAN Security: Cost-Effective Intrusion Detection for Real-Time Control Systems, SAE Technical Paper 2014-01-0340 (2014).
[5] Waszecki, P., Mundhenk, P., Steinhorst, S., Lukasiewycz, M., Karri, R. and Chakraborty, S.: Automotive Electrical and Electronic Architecture Security via Distributed In-Vehicle Traffic Monitoring, *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems* (2017).
[6] Taylor, A., Japkowicz, N. and Leblanc, S.: Frequency-based anomaly detection for the automotive CAN bus, *WCICSS2015* (2015).
[7] Lee, H., Jeong, S.H. and Kim, H.K.: OTIDS: A Novel Intrusion Detection System for In-vehicle Network by using Remote Frame, *Privacy, Security and Trust* (*PST*) (2017).
[8] Yajima, J., Abe, Y. and Hasebe, T.: Proposal of Anomaly Detection Method "Cumulative Sum Detection" for In-Vehicle Networks, *escar Asia 2018* (2018).
[9] Chen, Y., Kurachi, R., Zeng, G. and Takada, H.: Schedulability Comparison for CAN Message with Offset: Priority Queue Versus FIFO Queue, *19th International Conference on Real-Time and Network Systems* (Sep. 2011).
[10] Davis, R.I., Burns, A., Brill, R.J. and Lukkien, J.J.: Controller Area Network (CAN) schedulability analysis: Refuted, revisited and revised, Real-Time Systems (2007).
[11] Davis, R.I. and Navet, N.: Controller area network (CAN) schedulability analysis for messages with arbitrary deadlines in FIFO and work-conserving queues, *9th IEEE International Workshop on Factory Communication Systems* (2012).

Jun Yajima was born in 1974. He received his B.E. and M.E. in information and system engineering from Chuo University in 1997 and 1999, respectively. He is a senior researcher at Fujitsu Laboratories Ltd. He is also a Ph.D. student of the Institute of Information Security. His current research interest includes machine learning, information security, cryptography, and vehicle security. He was awarded the SCIS2007 paper prize and the SCIS2012 innovation paper prize.

Yasuhiko Abe was born in 1964. He received his B.E. in electrical and electronic engineering from Chuo University in 1989. He is a senior manager of Fujitsu Limited. His current research interest includes digital rights management and vehicle security.

Takayuki Hasebe was born in 1961. He received his B.E. and M.E. in electrical and electronic engineering from Tokyo Institute of Technology in 1983 and 1985, respectively. He is an expert researcher in Fujitsu Laboratories Ltd. His current research interest includes mobility system security.



Takao Okubo was born in 1966. He is a professor of Institute of Inforamtion Security, Japan. He received his M.S. degree in Engineering from Tokyo Institute of Technology in 1991. From 1991 to 2013 he worked as a researcher in Software Engineering and Software Security with Fujitsu Laboratories Ltd. He received the Ph.D. degree in Informatics from the Institute of Information Security in 2009. In 2013 he moved to the Institute of Information Security as an associate professor. He is a member of IEICE, IPSJ, ACM and IEEE CS. His current interests are secure development and threat analysis.