Exploring Attack Surfaces of Voltage-Based Intrusion Detection Systems in Controller Area Networks

Sang Uk Sagong, Xuhang Ying, Radha Poovendran, and Linda Bushnell
Department of Electrical and Computer Engineering, University of Washington, Seattle, WA 98195
Email: {sagong, xhying, rp3, lb2}@uw.edu

***Abstract*— Electronic Control Units (ECUs) in automobiles exchange information using in-vehicle network protocols such as the Controller Area Network (CAN). Designed for isolation, these protocols do not have security mechanisms such as message authentication or encryption. In order to secure the CAN protocol, anomaly-based Intrusion Detection Systems (IDSs) have been proposed to track physical properties and detect unexpected deviations from their normal behaviors. Voltage-based IDS (VIDS) exploits voltage characteristics for anomaly detection. To measure the voltage of the CAN bus, a VIDS requires additional wires to connect the microcontroller to the CAN bus. As a result, these wires may in turn introduce new attack surfaces to the CAN bus if the VIDS itself is compromised. In this paper, we propose three voltage-based attacks: 1) the overcurrent attack, in which the adversary damages the compromised ECU's microcontroller by letting the current that flows into an analog pin exceed the maximum amount that the microcontroller can absorb, 2) the denial-of-service attack, in which the adversary prevents any message from being transmitted by setting the CAN bus to an idle state, and 3) the forced retransmission attack, in which the adversary forces an ECU to retransmit by inducing an error during message exchange. To defend against the above attacks, we propose a hardware-based Intrusion Response System (IRS) that disconnects the VIDS from the CAN bus at the onset of the attacks. We demonstrate the proposed attacks on a CAN bus testbed and evaluate the effectiveness of the proposed IRS.**

***Index Terms*—Controller Area Network (CAN), Voltage-based Intrusion Detection System (IDS), Voltage-based Attack, Hardware-based Intrusion Response System (IRS)**

I. Introduction

Electronic Control Units (ECUs) in an automobile exchange information via in-vehicle network protocols such as Controller Area Network (CAN) [1], Local Interconnect Network (LIN) [2], and FlexRay [3]. Designed for closed networks, such in-vehicle network protocols do not have cryptography primitives such as message authentication code or encryption [1]–[3]. However, as more and more outward-facing ECUs (e.g., CD players, Wi-Fi/cellular/Bluetooth radios) are added to the network, the closed network assumption no longer holds, resulting in vulnerabilities to cyber attacks [4]–[8]. It is difficult to incorporate cryptographic primitives to existing in-vehicle protocols due to the backward compatibility with the legacy systems and the resource constraints of ECUs such as memory and computing power [9]. Hence, anomaly-based Intrusion Detection Systems (IDSs) have been proposed to detect suspicious behaviors on the CAN bus, which indicate the onset of the cyber attacks such as suspension, fabrication, masquerade, and bus-off attacks [10], [11]. Physical properties such as message periodicity [12], entropy [13], clock skew [10], [14], and voltage [9], [15], [16] are often exploited for developing IDSs.

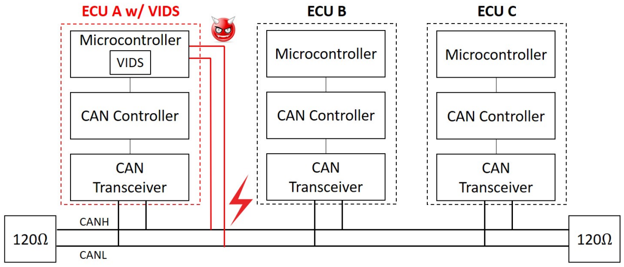


Fig. 1: General architecture of VIDS. VIDS is implemented as software code at ECU A's microcontroller. The microcontroller of ECU A is directly connected to the CAN bus via two wires (two red lines). If ECU A is compromised, the adversary can launch voltage-based attacks using the directly connected wires.

Among various physical properties, the voltage characteristics, such as voltage distribution [15] and transition time between bits 0 and 1 [16], of an ECU are more difficult to be mimicked by an adversary since they depend on the CAN transceiver's hardware. Hence, voltage-based IDSs (VIDSs) that exploit the voltage characteristics to fingerprint each ECU are more effective than the IDSs that rely on message periodicity and content [9], [15], [16]. As illustrated in Fig. 1, a VIDS is expected to be implemented as software code that resides in the microcontroller of an ECU equipped with a CAN controller and CAN transceiver. Although there exist two wires between the CAN bus and the CAN transceiver, they cannot be used by the VIDS for voltage measurements due to the following two reasons. First, they are solely used by the CAN transceiver for message exchange and not accessible by the microcontroller. Second, while the CAN transceiver can measure the differential voltage of the CAN bus, the CAN controller only provides the bit information to the microcontroller, which is not useful for the VIDS. Hence, it becomes necessary to introduce two additional wires to connect the microcontroller (the VIDS) with the CAN bus in order to collect voltage measurements, as demonstrated in [9], [15], [16].

These two extra wires open up the chance of vulnerabilities. Our observation is that if the VIDS is compromised, instead of passively measuring the voltage levels through the two wires, the adversary can actively manipulate the voltage levels applied to the analog pins in order to impede the CAN transceivers' operation and disable the VIDS through the two connected wires. In this paper, we explore possible attack surfaces of the VIDSs and propose three voltage-based attacks: 1) the overcurrent attack, 2) the denial-of-service (DoS) attack, and 3) the forced retransmission attack. In the overcurrent attack, the adversary manipulates the voltage of the CAN bus so that the current that flows into the microcontroller exceeds the hardware limit (i.e., the current absolute maximum rating), thus damaging the microcontroller. In the DoS attack, the adversary keeps the CAN bus in an idle state by holding the voltage of the CAN bus, thus preventing any messages from being transmitted. Hence, the DoS is observed by all other ECUs. In the forced retransmission attack, the adversary intentionally violates the bit timing requirement of the CAN protocol in order to generate an error during message transmission, resulting in message retransmission. We make following contributions:

- We propose three voltage-based attacks, i.e., the overcurrent attack, the DoS attack, and the forced retransmission attack against the VIDS and the CAN bus.
- We propose a hardware-based Intrusion Response System (IRS) based on fuses or circuit breakers that immediately disconnects the VIDS from the CAN bus at the onset of the proposed attacks.
- We demonstrate and evaluate the proposed voltage-based attacks and the proposed IRS on a CAN bus testbed. Our hardware evaluations show that the voltage-based attacks are feasible and practical. We also show that the IRS using the fuses can effectively mitigate the voltage-based attacks.

The rest of the paper is organized as follows. Section II reviews the related works on IDSs for the CAN protocol. Section III provides brief background on the CAN protocol and explains CAN transceiver operations. The adversary model is presented in Section IV, and the proposed voltage-based attacks are presented in Section V. The hardware-based IRS is described in Section VI. Section VII presents the experimental results. Section VIII concludes the paper.

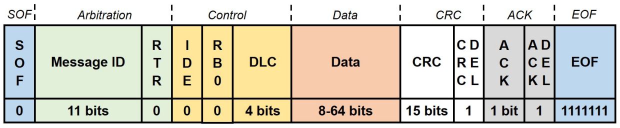
II. Related Work

Recent works have demonstrated that automobiles equipped with many outward-facing ECUs are vulnerable to the cyber attacks such as disabling brakes [4]–[8], [17] since the in-vehicle network protocols do not have cryptography primitives. Hence, due to the urgent need of securing the CAN protocol, many works have proposed various anomaly-based IDSs that detect the cyber attacks using abnormal deviations with respect to the traffic through the CAN bus. Since most of the messages in the CAN protocol are transmitted with a fixed length and frequency, the authors of [12] proposed an IDS that detects existence of spoofed messages using the frequency of the message occurrence. Also, the entropy-based IDS that exploits coincidence among a set of messages are proposed in [13]. However, an intelligent adversary can bypass the entropy-based IDS by replicating the structure and pattern of the legitimate traffic [10]. Hence, the IDS that exploits physical invariants of ECUs is proposed in order to detect the intelligent adversary, and the authors of [10] proposed the clock-based IDS (CIDS) that exploits clock skew of ECUs to fingerprint each ECU. However, the authors of [14] proposed the cloaking attack that bypasses the CIDS by matching the interarrival time of the spoofed messages to that of the legitimate messages.

Compared with physical properties like message periodicity, traffic pattern, and clock skew, the voltage characteristics of an ECU are more difficult to be mimicked, since the voltage characteristics are determined by the CAN transceiver's hardware such as internal impedance of the transistors and diodes which are not affected by the software of the microcontroller [15]. Despite a potential risk of connecting the microcontroller's analog pins to the CAN bus, the VIDSs are proposed in recent works [9], [15], [16]. For instance, in [9], the mean squared error between the measured voltage and the reference voltage of each ECU that has been collected before the attack is exploited for developing an IDS. In [16], the proposed VoltageIDS exploits the voltage difference between the two CAN bus lines and transition time between bits 0 and 1 in order to detect the deviation from the normal voltage behavior using machine learning algorithms. In [15], Viden is proposed that measures the voltages of each CAN bus line and extracts features from the distribution of the measured voltage samples.

The VIDSs have to measure the voltages of the CAN bus lines in order to extract the voltage characteristics of each ECU. In [9] and [16], the voltage is measured using an oscilloscope that may not be a viable option for automobiles due to its required power and space. Alternatively, a VIDS may be implemented in a microcontroller like the Viden [15]. Since a microcontroller requires typically 7-12V that can be supplied from the car battery, a microcontroller can be operated at an automobile. Hence, the Viden measures the voltage and detects an attack in more practical environment such as driving an automobile although the voltage is measured with slower rate and less accuracy compared to an oscilloscope. If the microcontroller or oscilloscope is compromised, the adversary may exploit the wires connected to the CAN bus lines to launch the attacks. However, the recent works on the VIDSs did not discuss any potential cyber attacks in which these wires are maliciously used, nor did the works propose a protection or security mechanism to mitigate the attacks [9], [15], [16].

Although it is demonstrated that an IDS is effective in protecting the CAN bus, the IDS is limited to detecting cyber attacks and alerting an operator of the automobile. The reaction to the detected attacks remains to the operator of the automobile [18]. However, after detecting the attacks, an



SOF	Arbitration		Control			Data	CRC		ACK		EOF
SOF	Message ID	RTR	IDE	RB0	DLC	Data	CRC	CRC DEL	ACK	ACK DEL	EOF
0	11 bits	0	0	0	4 bits	8-64 bits	15 bits	1	1 bit	1	1111111

Fig. 2: Structure of a data frame in the CAN protocol.

IRS can promptly minimize the consequence of the attacks or remove the attacks [12], [18].

III. Preliminaries

In this section, we review the CAN protocol and explain transistor operations in CAN transceivers. Then, we discuss analog pin settings at the microcontrollers for measuring the voltage and generating electric signals.

A. CAN Protocol Background

The CAN protocol is *de facto* one of the most widely used in-vehicle network protocols [1], [19]. As a multi-master broadcast medium, the CAN bus allows ECUs to transmit messages and observe all ongoing messages. In the CAN protocol, messages are transmitted on the first come, first served basis. If two messages are transmitted at the same time, the message with a smaller ID (higher priority) will be transmitted through a non-destructive content-based process called *arbitration*. For instance, suppose that ECUs A and B attempt to simultaneously transmit their messages with IDs 0x110 and 0x001, respectively. Since the CAN bus acts as a logical AND gate, ECU A would observe a bit 0 although it had transmitted a bit 1. Hence, ECU A realizes that ECU B is transmitting a higher priority message and stops transmission. Therefore, bits 0 and 1 are called as *dominant* bit and *recessive* bit, respectively.

Fig. 2 illustrates the structure of a CAN data frame. There are a total of 7 fields, out of which only the data field has a variable length (1-8 bytes). Note that there are no fields assigned for encryption or authentication in the data frame as well as in all other types of frames including remote, error, and overload frames. Due to external electrical/physical interference or malfunction of the CAN transceivers, a message may not be transmitted. If the message transmission fails, the message is retransmitted followed by an error frame.

B. Operation of CAN Transceivers

The CAN bus is composed of two bus lines called CANH and CANL, terminated by two 120Ω resistors as shown in Fig. 1. Since the CAN protocol uses the differential voltage to represent a bit, it is more robust to the external electromagnetic interference [20], [21]. When a dominant bit is transmitted, the differential voltage becomes larger than a predetermined threshold, typically 0.9V, and it remains 0.0V for a recessive bit.

Fig. 3 shows the voltages of CANH and CANL when dominant and recessive bits are transmitted by a CAN transceiver with 5V supply voltage [22], [23]. When a dominant bit is transmitted, CANH and CANL become nominal 3.5V and 1.5V, respectively. Hence, the differential voltage becomes 2.0V. However, when a recessive bit is transmitted, the differential voltage is 0.0V since both CANH and CANL are set to nominal 2.5V.[1]

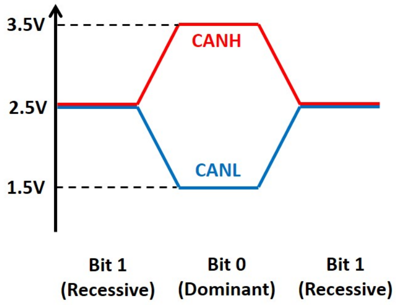


Fig. 3: Voltage levels of CANH and CANL when dominant and recessive bits are transmitted by a 5V CAN transceiver.

The voltages of the CAN bus are controlled by two transistors in a CAN transceiver as shown in Fig. 4. When transmitting a recessive bit, both transistors are in the off-state. Hence, the current cannot flow from the supply voltage (V_{DD}) to the ground. Since the reference voltage (e.g., nominal 2.5V in a 5V CAN transceiver) is applied to both CANH and CANL, CANH and CANL are set to the reference voltage. Hence, the differential voltage becomes 0.0V. Since the electric potential difference across the termination resistors is 0.0V, the current does not flow through the termination resistors.

When transmitting a dominant bit, the driver control sets both transistors in the on-state. Hence, the electric path is created between the emitter and the collector in bipolar junction transistors (BJTs) or the source and the drain in field effect transistors (FETs). Then, the current flows from the supply voltage to the ground. Due to the current flowing through the termination resistors, a voltage drop is induced between CANH and CANL. As a result, the differential voltage becomes non-zero. Since the operations of BJT and FET as switches are the same, the analysis on the operation of Microchip MCP2551 CAN transceivers can be applied to any CAN transceivers that are composed of FETs [23]. Then, we explain how the voltages of CANH and CANL are set to 3.5V and 1.5V, respectively. Since a Microchip MCP2551 CAN transceiver is operated by 5.0V, V_{DD} is set to 5.0V. When the BJT is in the on-state, the voltage is dropped by 0.7V between the base and the emitter of the BJT. Hence, the voltage at the collector becomes 4.3V. Since the voltage is again dropped by 0.7V at the diode from 4.3V, the voltage of CANH becomes 3.5-3.6V. Similarly, the voltage of CANL becomes 1.4-1.5V since the voltage is increased by 0.7V at

[1] A 3.3V CAN transceiver sets CANH and CANL to 2.3V for a recessive bit [24], [25]. When transmitting a dominant bit, CANH and CANL are set to 3.0V and 1.0V, respectively. Although the nominal voltages of the CAN bus may be different depending on the types of the CAN transceivers, the differential voltage between CANH and CANL is used for a bit representation in all CAN transceivers [20], [22]–[26]. Hence, throughout this paper, we consider 5V CAN transceivers.

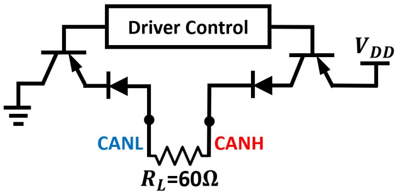


Fig. 4: The schematics of the Microchip MCP2551 CAN transceiver, which is composed of two BJTs and two diodes. The driver control turns off the BJTs when transmitting a recessive bit and turns them on when transmitting a dominant bit. The impedance R_L between CANH and CANL is 60Ω since two 120Ω termination resistors are connected in parallel.

the BJT and the diode connected to CANL, respectively.

C. Analog Pin Allocation in Microprocessors

Most of the microprocessors can measure the voltage using their analog pins since Analog-to-Digital Converters (ADCs) are implemented inside the microprocessors [27]–[31]. An analog pin has to be in the input mode when that analog pin is used for measuring the voltage. In addition to the input mode, an analog pin can generate electric signals such as a Pulse Width Modulation (PWM) signal if the analog pin is set to the output mode [27]–[29]. Since the analog pin mode is selected by the microcontroller's software, an adversary may change an analog pin from the input mode to the output mode by uploading its malicious code to the microcontroller via the cyber attacks without modifying any physical component of an automobile. For instance, a Microchip ATmega328P used in the Arduino UNO Rev3 board, NXP MPC563 used in an ECU of Hyundai Sonata, and Renesas V850 used in an Engine Control Module (ECM) of Toyota Camry provide analog pins that can be used to measure the voltage or generate PWM signals [28], [30], [31].

IV. System and Adversary Models

In this section, we present our system model and describe our adversary model for the voltage-based attacks on the VIDS and the CAN bus.

A. System Model

We consider a VIDS that is implemented as software code running on an ECU's microcontroller as shown in Fig. 1. The VIDS may be compromised in the following ways. First of all, if an adversary has physical access to the vehicle, it can exploit the on-board diagnostics (OBD-II) port that is mandated for all automobiles in EU and US [32], [33] to access the CAN bus and upload malicious code to a targeted ECU through a pass-thru device such as Hyundai Global Diagnostic System [34] and Volkswagen VAG-COM Diagnostic System [35]. It has also been demonstrated in [8] that an adversary is able to upload malicious code to any ECUs through a compromised ECU.

It is also possible for an adversary to remotely compromise an ECU without physical access to the vehicle as demonstrated in [4], [8], [36]. We consider two cases: 1) the adversary compromises the VIDS that is implemented on an ECU having a telematics unit, and 2) the adversary first compromises an ECU with a telematics unit and then compromises the VIDS using the compromised ECU. In the first case, the adversary can remotely access to the operating system of the ECU having a telematics unit to figure out the particular code that handles wireless connectivity such as Bluetooth. By exploiting that particular code, the adversary may execute its malicious code on that ECU. Hence, the VIDS can be compromised if it is implemented on the ECU having a telematics unit [8]. In the second case, the adversary can use the remotely compromised ECU with a telematics unit to upload its malicious code to the targeted ECU.

B. Adversary Model

While an adversary can compromise the ECU that implements the VIDS, it cannot modify the firmware of the CAN controller without the required special equipment [37]. Hence, the compromised ECU cannot manipulate the CAN transceivers' output voltages to violate the CAN protocol. However, once an ECU with the VIDS is compromised, the adversary can take the full control of the microcontroller and manipulate analog pin settings. Hence, the adversary can apply electric signals to CAN bus by assigning different modes to the analog pins as explained in Section III.

V. Voltage-based Attacks

In this section, we describe the analog pin settings for the VIDS and propose three voltage-based attacks, namely, the overcurrent attack, the DoS attack, and the forced retransmission attack against the VIDS and the CAN bus.

A. Voltage Manipulation of Analog Pins

When the two wires are introduced to connect the microcontroller and the CAN bus, they are expected to be soldered to two analog pins on the microcontroller. Let P_H and P_L denote the analog pins connected to CANH and CANL, respectively. P_H and P_L can operate in one of the following three modes: 1) input mode, 2) high voltage output mode (5.0V), and 3) low voltage output mode (0.0V). Note that there are only two output voltage levels, because the analog pin can only be turned on or off, when in the output mode.[2]

During the normal operations of the VIDS, the analog pins operate in the input mode, which allows the VIDS to passively measure the voltages of CANH and CANL. However, if the VIDS is compromised, the adversary can set P_H and P_L to arbitrary modes, some of which can cause damage to the microcontroller or impede the operations of the CAN transceivers. Table I lists all the 9 combinations of the operating modes of the two analog pins, and some may result in three voltage-based attacks, as explained in the rest of this section.

[2] The output voltage level from a microcontroller can be binary voltage levels (either high or low voltage levels) or multiple voltage levels depending on the microcontroller [27]. In this paper, we consider that the binary output voltage levels.

B. Overcurrent Attack

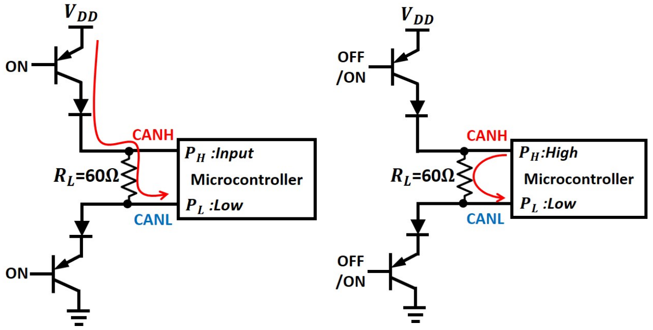
The overcurrent attack occurs when P_H is set to the high voltage output mode or input mode and P_L is set to the low voltage output mode. The idea of this attack is to make the current that flows into the analog pin of the microcontroller exceed the absolute maximum rating of the microprocessor (i.e., the current limit I_{max} that the microprocessor can absorb.), thus rendering the microprocessor to malfunction or get burned due to the electric shock. For instance, the values of I_{max} are 40mA for Microchip ATmega328P [27] and Renesas V850 [38] and 20mA for NXP MPC563 [31].[3]

When the microcontroller measures the voltage, the current does not flow through an analog pin of the microcontroller ideally. However, a microcontroller draws a very small current for measuring the voltage in practice, and the current depends on the accuracy of voltage measurements. This current is negligible due to the high impedance at the microcontroller. Hence, the current that flows through an analog pin in the voltage measurement is typically less than I_{max}, not damaging the microcontroller.

On the other hand, the adversary may make the current flow into an analog pin by manipulating the pin mode as illustrated in Fig 5. Let us denote $V_{H,b}$ and $V_{L,b}$ as the voltages of CANH and CANL when bit b is transmitted, respectively, where b is either 0 or 1. For example, $V_{H,0}$ indicates the voltage of CANH when a dominant bit is transmitted. Using Fig. 5, the condition of $V_{H,0} - V_{L,0}$ for the overcurrent attack can be derived in terms of R_L and I_{max} as follows

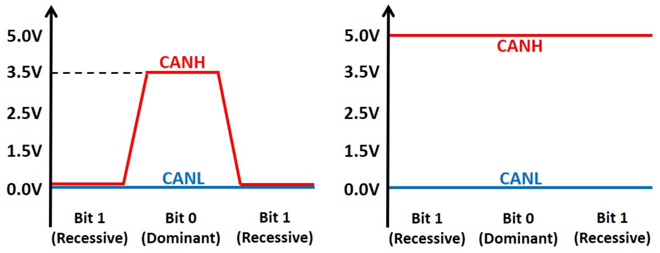
$$\frac{(V_{H,0} - V_{L,0})}{R_L} > I_{max}, \tag{1}$$

where R_L is 60Ω and I_{max} depends on a hardware limit of the microcontroller. Depending on the voltage of $V_{H,0}$ with $V_{L,0}$=0.0V, we propose two types of the overcurrent attacks, namely, the passive and the active overcurrent attacks.



(a) Passive overcurrent attack (b) Active overcurrent attack

Fig. 5: Circuit diagrams under the overcurrent attacks. The red curve indicates the current in each overcurrent attack. (a) In the passive overcurrent attack, the current flows from V_{DD} to P_L when a dominant bit is transmitted. (b) In the active overcurrent attack, the current flows from P_H to P_L regardless of the bits.



(a) Passive overcurrent attack (b) Active overcurrent attack

Fig. 6: Voltage behavior of CANH and CANL under the overcurrent attacks. (a) In the passive overcurrent attack, the maximum voltage difference between CANH and CANL is 3.5V when a dominant bit is transmitted. (b) In the active overcurrent attack, the voltage difference is always 5.0V.

In the passive overcurrent attack, the adversary changes P_L from the input mode to the low voltage output mode, as illustrated in Fig. 5(a). Since $V_{H,0}$ is 3.5V, the current that flows into the analog pin is computed as $\frac{3.5V-0.0V}{60\Omega}$=58.3mA, as illustrated in Fig. 6(a). In the passive overcurrent attack, the current that flows into the microcontroller is supplied from the car battery that can supply the current in the order of Ampere. In the active overcurrent attack, the adversary changes P_H and P_L from the input mode to the high voltage output mode and the low voltage output mode, respectively, as illustrated in Fig. 5(b). Then, as shown in Fig. 6(b), the current that flows into the analog pin becomes $\frac{5.0V-0.0V}{60\Omega}$=83.3mA, which is supplied from the microcontroller. As shown in Table I, if either P_H or P_L is set to the low voltage output, the overcurrent attack is launched since the current flows into that analog pin.

TABLE I: Combinations of operating modes of P_H and P_L.

P_H mode	P_L mode	Is it an attack?
Input	Input	Not an attack, setting for measuring voltage
Input	High	DoS attack
Input	Low	Passive overcurrent attack
High	Input	Forced retransmission attack
High	High	Not valid setting
High	Low	Active overcurrent attack
Low	Input	DoS attack or passive overcurrent attack
Low	High	DoS attack or active overcurrent attack
Low	Low	DoS attack or passive overcurrent attack

C. Denial-of-Service (DoS) Attack

The DoS attack occurs when P_L is set to the high voltage output mode while P_H is in the input mode. The idea of this attack is to increase the voltage level of CANL such that the differential voltage V_{Diff} between CANH and CANL drops below the decision threshold for determining the dominant bit. Hence, the CAN bus observes a recessive bit while an ECU tries to transmit a dominant bit, and messages cannot be transmitted through the CAN bus. Let us denote $V_{attack,L}$

[3] The datasheet of Renesas V850 does not provide the current absolute maximum rating. Since the voltage absolute maximum rating of Renesas V850 and operating condition such as temperature are similar to that of Microchip ATmega328P, it is reasonable to assume that the current absolute maximum rating of Renesas V850 is similar to Microchip ATmega328P which is 40mA.

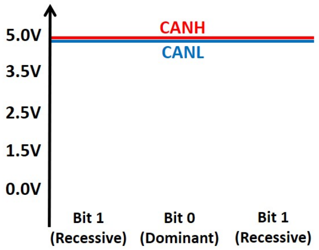


Fig. 7: Voltages of CAN bus under the DoS attack. Since $V_{attack,L}$ is set to 5.0V, both CANH and CANL become 5.0V, which represents a recessive bit.

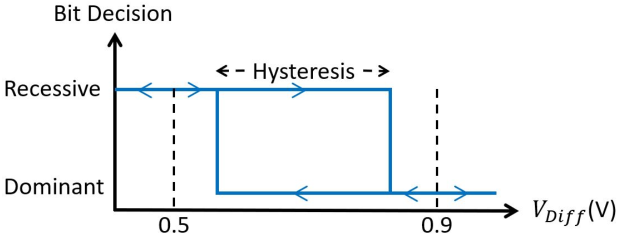


Fig. 8: Bit decision criteria of the Microchip MCP2551 CAN transceiver. The differential input hysteresis may not be identical among the CAN transceivers. A CAN transceiver can determine a dominant bit if V_{Diff} is typically larger than 0.9V and a recessive bit if V_{Diff} is less than 0.5V.

as the voltage applied to CANL by the adversary using P_L. Then, since the voltage of CANL is manipulated to $V_{attack,L}$, V_{Diff} can be computed as follows

$$V_{Diff} = V_{H,b} - V_{attack,L}. \tag{2}$$

When transmitting a recessive bit, $V_{H,1}$ is set to $V_{attack,L}$ since the voltages of CANH and CANL are the same due to no current through the termination resistors. Hence, using Eq. (2), V_{Diff} becomes 0.0V, representing a recessive bit even though the voltages of CAN bus are manipulated as illustrated in Fig. 7. When an ECU tries to transmit a dominant bit, the adversary may make V_{Diff} less than the decision threshold for determining the dominant bit by increasing $V_{attack,L}$ according to Eq. (2). As illustrated in Fig. 8, a CAN transceiver can only decide a dominant bit if 0.9V$<V_{Diff}<$5.0V with the differential input hysteresis between 100-200mV. Hence, any value of $V_{attack,L}$ that makes V_{Diff} <0.9V can successfully thwart the transmission of a dominant bit. When $V_{attack,L}$ is greater than 3.5V, $V_{H,0}$ is increased to $V_{attack,L}$ since the current cannot flow through the termination resistors due to the diode in the CAN transceiver. Hence, V_{Diff} becomes 0.0V as shown in Fig. 7, and the DoS attack becomes successful. The DoS attack can be also successfully launched if P_H is set to the low voltage output mode regardless of the mode of P_L since V_{Diff} becomes less than or equal to 0.0V which does not meet the decision threshold for determining the dominant bit as shown in Fig. 8.

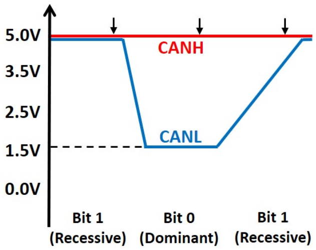


Fig. 9: Voltages of CAN bus under the forced retransmission attack when $V_{attack,H}$ is set to 5.0V. When a recessive bit is transmitted right after a dominant bit, the transition time of the CANL's voltage level increases. Hence, a receiving ECU may determine a dominant bit since V_{Diff} is greater than 0.9V when it samples the CAN bus at the instances indicated with the black arrows facing downward.

D. *Forced Retransmission Attack*

The forced retransmission attack occurs when P_H is set to the high voltage output mode and P_L is set to the input mode. The idea of this attack is making the transition time from a dominant bit to a recessive bit longer than the nominal transition time whose typical value is 70-130ns [22]. By doing so, the adversary can induce an error in the message reception, especially at the ACK delimiter position which has to be the recessive bit as shown in Fig. 2. In order to let the CAN bus represent a dominant bit at the ACK delimiter position, the adversary increases the transition time. Due to the increment of the transition time, the CAN bus may represent a dominant bit as illustrated in Fig. 9.

In order to quantitatively compare the transition time from a dominant bit to a recessive bit in the normal transmission and under the attack, we define the *bit length time* τ_{bit} as follows

$$\tau_{bit} \triangleq t_{90\%} - t_{start\ of\ bit\ 0}, \tag{3}$$

where $t_{start\ of\ bit\ 0}$ and $t_{90\%}$ denote the time at which a dominant bit starts and the time at which the voltage of CANL reaches 90% of $V_{attack,H}$, respectively as illustrated in Fig. 10, which is the sum of the duration of the dominant bit and the transition time. This definition of τ_{bit} is reasonable since the rise time and fall time in many RLC circuits are defined in the same way [39]. For instance, if the CAN bus speed is set to 500kbps, τ_{bit} without the attack is nominal 2μs.

Let us denote $V_{attack,H}$ as the applied voltage to CANH by the adversary using P_H. Since the nominal voltage of CANH in the idle state is 2.5V, $V_{attack,H}$ greater than 2.5V is considered. When a recessive bit is transmitted, both $V_{H,1}$ and $V_{L,1}$ become $V_{attack,H}$ since the current does not flow through the termination resistors. Hence, V_{Diff} is 0.0V which represents a recessive bit. When transmitting a dominant bit, V_{Diff} increases as increasing $V_{attack,H}$ since $V_{L,0}$ stays at 1.4-1.5V. For $V_{attack,H}$ >3.5V, the adversary forcefully sets $V_{H,0}$ higher than the voltage in the normal operation of a CAN transceiver. Hence, the voltage of CANL has to be pulled up to $V_{attack,H}$ from 1.4-1.5V when a recessive bit is transmitted

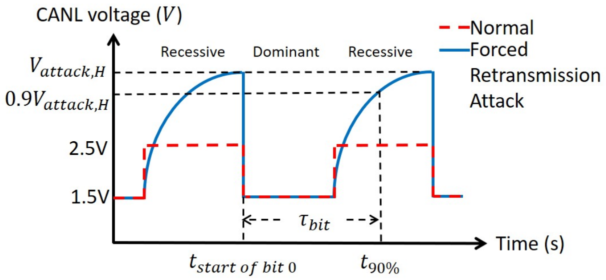


Fig. 10: Voltage of CANL under the forced retransmission attack. The bit length time τ_{bit} is defined as the sum of duration of the dominant bit and the transition time. Compared with the voltage characteristics of the normal operation (dashed red line), the voltage of CANL needs to be pulled up to $V_{attack,H}$ and the transition time increases (solid blue line).

right after a dominant bit. As increasing $V_{attack,H}$, $t_{90\%}$ also increases, and the transition time τ_{bit} increases as illustrated in Fig. 10.

The CAN transceivers of the receiving ECUs sample the voltages of the CAN bus lines at the fixed period based on their local clocks according to the CAN bus speed. Hence, the CAN transceivers cannot decode a transmitted bit correctly if the duration of a bit in a message violates the CAN protocol. As a result, an error occurs while receiving a message, and the transmitting ECU sends that message again.

VI. Hardware-based Intrusion Response System

In this section, we introduce our hardware-based IRS and explain in detail how the proposed IRS can protect the CAN bus from the voltage-based attacks.

A. Proposed IRS

As shown in Fig. 11, the proposed IRS consists of fuses or circuit breakers that are attached between the microcontroller's analog pins and the CAN bus. Whenever the voltage-based attacks occur, the proposed IRS will immediately disconnect the VIDS physically from the CAN bus, thus avoiding any damage to the CAN bus or the microcontroller. Since the current does not flow through the analog pins of the microcontroller in the normal operation of the VIDS, the fuses or circuit breakers may exploit the current flow through these analog pins as an indicator of the voltage-based attacks. These hardware circuit components such as the fuses or circuit breakers operate independently to the cyber component of an ECU, especially the software of the microcontroller. Hence, the proposed IRS cannot be disabled by the cyber attacks on the automobile. In the rest of this section, we will explain how the proposed IRS can mitigate the proposed voltage-based attacks.

B. Overcurrent Attack

In order to protect a microcontroller from the overcurrent attack, the current that flows into the microcontroller of the compromised ECU has to be limited to be smaller than the absolute maximum rating I_{max}. Hence, a current limiting circuit, a fuse, or a circuit breaker can be used to limit the current. Although the current limiting circuit, which is one of the most straightforward solutions for limiting the current, can protect the microcontroller from being damaged by the overcurrent attack, the microcontroller cannot measure the voltage with the current limiting circuit due to the load resistors in the current limiting circuit. Since the VIDS cannot operate properly, the current limiting circuit cannot be implemented as the IRS to the overcurrent attack.

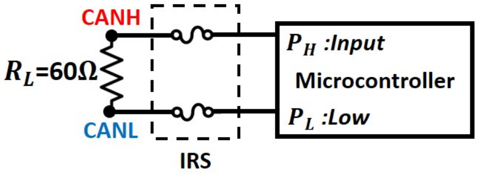


Fig. 11: Structure of the proposed hardware-based IRS that comprises two fuses connected to each analog pin of the microcontroller.

A fuse can be used to limit the current since it disconnects two parts of the circuit when the current higher than its current rating flows longer than its opening time. The typical opening time of the very fast-acting fuses is in the order of μs to ms, and the fuse opens faster if the current rating becomes smaller and the actual current that flows through the fuse increases [40], [41]. Since it takes longer time to damage a microcontroller by the overcurrent than to open a fuse [27], the fuse disconnects the wire before the microcontroller is burned. In case of the microcontrollers that can be damaged by the overcurrent within a short amount of time which is between ns and μs, we can use micro-electro mechanical system (MEMS)-based fuses or thin film-based fuses whose opening times are in the order of μs for the MEMS-based fuse [42] and in the order of ns for the thin-film based fuse [43]. Moreover, the fuse does not thwart the voltage measurement at the microcontroller when the fuse is attached to an analog pin as illustrated in Fig. 11 since the fuse is basically a wire which does not induce any voltage drop ideally. Also, implementing the fuse on a stock ECU is feasible with low cost because most of the automobiles are equipped with fuse boxes. Since the current that flows into P_L is computed as 58.3mA for the passive overcurrent attack and 83.3mA for the active overcurrent attack as mentioned in Section V, any fuses with the current rating less than 58.3mA can be used to protect the microcontroller from the overcurrent attack.

Although fuses have to be replaced every time they blow out, the replacement process is simple and fuses are cheap. Hence, replacing fuses is not a significant issue. However, circuit breakers are reusable after they disconnect the wires due to the overcurrent. In addition, since there are many fuses and circuit breakers with various values of the current ratings in the market, the most appropriate fuses and circuit breakers can be selected even though we cannot tune their current ratings. Between a fuse and a circuit breaker, the fuse is

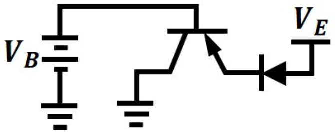


Fig. 12: Test circuits for measuring current in the DoS attack.

the best component for the proposed IRS, especially for the automobile since the fuse is lighter, smaller, and cheaper than the circuit breaker.

C. DoS Attack and Forced Retransmission Attack

In order to mitigate the DoS attack and the forced retransmission attack, the voltage from the analog pin has to be limited since these attacks are launched by applying abnormal voltage to the CAN bus. Hence, one may think of installing a voltage limiting circuit as a defense. Nevertheless, the microcontroller may not measure the voltage, which is the essential functionality of the VIDS, with the voltage limiting circuit since the Zener diodes admit the current nonlinearly according to the voltage. Therefore, the voltage limiting circuit cannot be implemented as the IRS to these attacks.

The fuses or circuit breakers can be used to mitigate the DoS attack and the forced retransmission attack since the current flows through the analog pins under these attacks. Under the DoS attack, the current flows through P_L since the electric path is created from P_L to the ground of the CAN transceiver when a dominant bit is transmitted. Also, the current flows through P_H under the forced retransmission attack because the electric path is created from the P_H to the ground of the CAN transceiver. Hence, using the current that flows through the analog pins as an indicator of these attacks, the fuses or circuit breakers can be implemented as the IRS to these attacks.

For determining the current rating of the fuse, we measure the current that flows through P_L and P_H under the DoS attack and the forced retransmission attack, respectively. For the DoS attack, we implement a test circuit that emulates the voltage characteristics of the transistors in the CAN transceiver as shown in Fig. 12 using a Motorola Solutions 2N2905A PNP BJT. The measured current that flows to the ground of the transistor is 281mA. For the forced retransmission attack, the current that flows through P_H can be computed as $\frac{(5.0\mathrm{V}-1.5\mathrm{V})}{60\Omega} = 58.3\mathrm{mA}$. Hence, any fuses or circuit breakers with the current rating less than 58.3mA may mitigate these attacks.

VII. Evaluation

In this section, we demonstrate the overcurrent attack, the DoS attack, and the forced retransmission attack on the CAN bus testbed. Moreover, we demonstrate that the proposed IRS is effective in defending the CAN bus against the proposed voltage-based attacks.

A. Testbed and Equipment

As shown in Fig. 13, the CAN bus testbed consists of three testbed ECUs. Each testbed ECU is composed of an Arduino UNO Rev3 board and a Sparkfun CAN bus shield. The CAN bus shield uses a Microchip MCP2515 CAN controller and a Microchip MCP2551 CAN transceiver. The two bus lines of the CAN bus testbed are terminated by two 120Ω resistors. The CAN bus speed is set to 500kbps that is widely used in many automobiles [10]. Since the VIDS is implemented at the microcontroller of ECU A, two analog pins (pin numbers A0 and A5) of the Arduino board in ECU A are connected to CANH and CANL, respectively. ECU A is compromised and launches the proposed voltage-based attacks by exploiting these analog pins. ECU C transmits messages every 1 second (i.e., 1Hz.). ECU B is a receiving ECU that transmits an acknowledgment bit for each properly received messages and logs all messages.

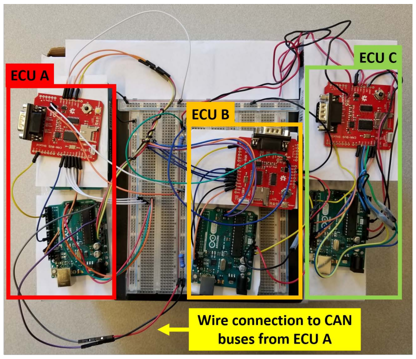


Fig. 13: The CAN bus testbed. The microcontroller of ECU A is connected to CANH and CANL via two wires. ECU A is compromised and launches the proposed voltage-based attacks. ECU C transmits messages every 1 second, and ECU B logs the received messages.

In order to understand the impact of the voltage-based attacks on the CAN bus better, we use an oscilloscope (Tektronix TDS2004B, up to 1GSamples/sec) to measure the voltage levels of the CAN bus lines and bit durations of messages. Since the Arduino board can only provide fixed voltages from the analog pins, we use a power supply (Keysight U8031A) to control the voltage levels of $V_{attack,H}$ and $V_{attack,L}$ in order to determine the minimum voltage levels for launching the DoS attack and the forced retransmission attack. A digital multimeter (Keysight 34461A) is used to measure the voltage and the current.

We use Littelfuse 0326 fuses whose current rating is 10mA to implement the proposed IRS. In order to check that the proposed IRS based on the fuses does not thwart the voltage measurement at the microcontroller, we implement a test circuit using two fuses as illustrated in Fig. 14, where V_{DD} is set to 5.0V. When the analog pins are in the input mode, the VIDS can correctly measure the voltage levels of CANH and CANL (5.0V and 0.0V at A0 and A5, respectively). We repeat the same experiment after replacing the fuses with the circuit breakers and check that the Arduino board can measure the voltage with the circuit breakers.

Fig. 15 shows the voltages of the CAN bus when a message

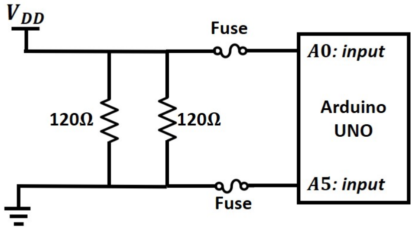


Fig. 14: Test circuits for checking that the Arduino board can measure the voltage with the fuses.

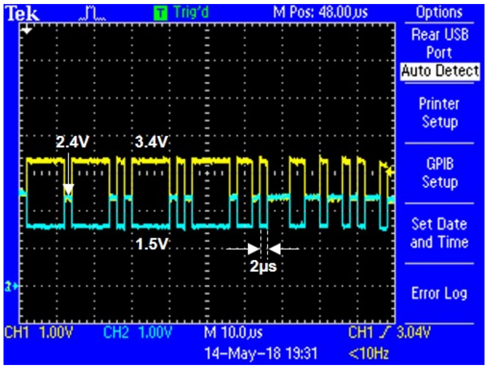


Fig. 15: Voltages of CANH (in yellow) and CANL (in blue) during the normal message transmission.

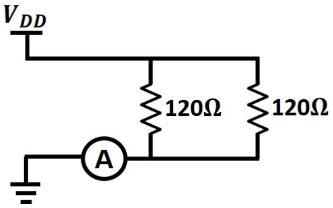


Fig. 16: Test circuit to emulate the voltages of the CAN bus under the overcurrent attacks. We measure the current that flows to the ground, and V_{DD} is set to 3.5V in the passive overcurrent attack and 5.0V in the active overcurrent attack.

is transmitted in the absence of the voltage-based attacks. Since the CAN bus speed is set to 500kbps, the nominal bit length time is 2μs. The actual voltages of CANH and CANL are 2.4V when a recessive bit is transmitted. When a dominant bit is transmitted, the actual voltages of CANH and CANL are 3.4V and 1.5V, respectively.

B. Overcurrent Attack

In order to avoid any damage to the testbed ECUs, we implement a test circuit as illustrated in Fig. 16 to emulate the CAN bus under the overcurrent attack. In the test circuit, the ground represents P_L of the microcontroller while the node with V_{DD} represents the voltage when a dominant bit is transmitted in the passive overcurrent attack and P_H of the microcontroller in the active overcurrent attack.

In order to emulate the passive overcurrent attack, V_{DD} is set to 3.5V using the power supply, and the measured current that flows into the ground is 58mA using the digital multimeter. When emulating the active overcurrent attack, V_{DD} is set to 5.0V, and the measured current flowing into the ground is 83mA. The experimental values of the current match with the theoretical values computed in Section V. Moreover, since the current is supplied from the microcontroller in the active overcurrent attack, the microcontroller may not generate 83mA. Hence, we measure that the maximum 52mA can be generated from an analog pin of the Arduino board, which is large enough to damage a majority of microprocessors, including Microchip ATmega328P, Renesas V850, NXP MPC563 [27], [31], [38].

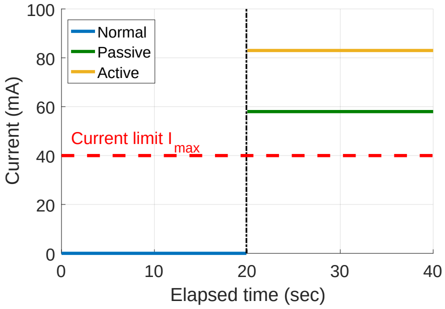


Fig. 17: Current measured in the test circuit under the overcurrent attacks. Before the attacks occur, the current is 0mA. However, after the attacks, the current greater than I_{max} (i.e., 40mA.) flows to the ground in both attacks which may damage the microcontroller.

Fig. 17 demonstrates the current flowing into the ground in the test circuit under the overcurrent attacks. The overcurrent attacks occur at t=20 second which is indicated by the black dashed line. The red dashed line indicates the current limit I_{max} of the Arduino board which is 40mA. Before the overcurrent attacks, the current does not flow to the ground. However, after the overcurrent attacks occur, the current suddenly increases above the red dashed line which means that the microcontroller can be damaged by both the passive and active overcurrent attacks. Since the current that is greater than 10mA flows under the overcurrent attacks, the fuse whose current rating is 10mA can be opened due to the attacks. Hence, the fuse can protect the microcontroller from the overcurrent attacks by disconnecting P_L from CANL.

C. DoS Attack

In order to demonstrate that the DoS attack is indeed feasible in the practical settings at an automobile, we design a DoS attack scenario using the CAN bus testbed. Initially, P_L of ECU A's Arduino board is set to the input mode for 11 seconds while the message is transmitted from ECU C every 1 second and logged at ECU B. Then, the DoS attack is launched using ECU A by setting P_L to the high voltage output mode that applies 5.0V to CANL. ECU A launches the DoS attack for 20 seconds and stops the attack by setting P_L back to the input mode. Using the oscilloscope, the voltages of each CAN bus line are observed under the attack.

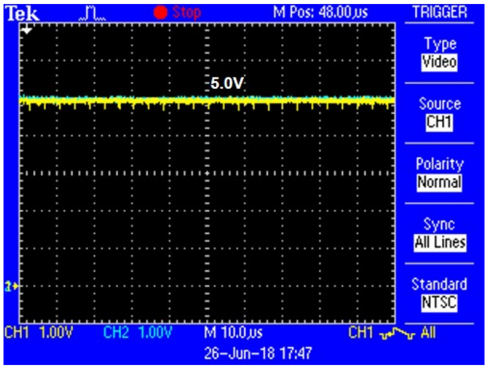
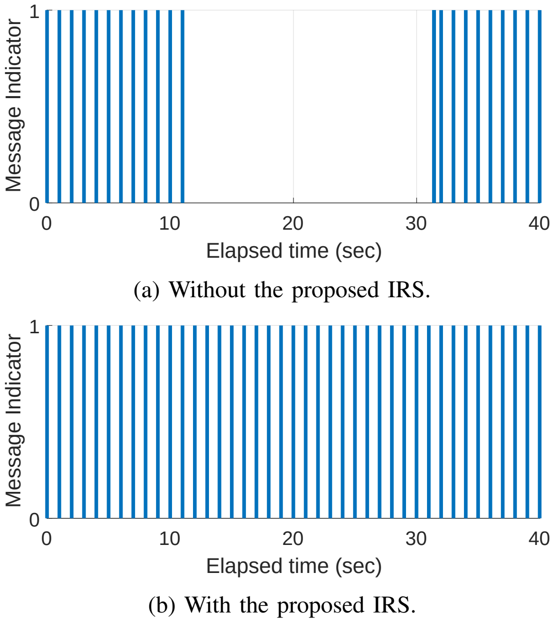


Fig. 18: Voltages of the CAN bus when the DoS attack is successfully launched by ECU A. P_L is set to the high voltage output mode that can generate 5.0V.

As shown in Fig. 18, the voltages of CANH and CANL become 5.0V since $V_{attack,L}$ is set to 5.0V. Since V_{Diff} is always 0.0V, a dominant bit cannot be transmitted. Hence, the DoS attack is successfully launched using ECU A. Fig. 19(a) demonstrates the message exchange between ECUs B and C when the DoS attack is launched by ECU A according to the attack scenario. When a message is received at t_0, the message indicator becomes 1 at t_0, and it is 0 if a message is not received. The message is received at ECU B every 1 second before the attack occurs at t=11 second. However, the messages are not exchanged between 11-31 seconds due to the attack. As soon as ECU A stops the attack at t=31 second, the CAN bus returns to the normal state, and the messages are exchanged again. One thing to note is that the interarrival time between the first two messages right after stopping the attack is shorter than 1 second since the CAN transceiver transmits the message that was saved in its buffer due to the transmission failure during the attack.

In order to demonstrate that the proposed IRS can mitigate the DoS attack, we implement the proposed IRS at ECU A, and ECU A launches the DoS attack at t=11 second as before. As shown in Fig. 19(b), the messages are exchanged between ECUs B and C every 1 second as normal during the attack. Hence, the proposed IRS can successfully mitigate the DoS attack.

We further investigate the DoS attack to determine the minimum voltage level that leads to a successful attack. The power supply is connected to CANL since the analog pin of the Arduino board can only generate either 0.0V or 5.0V. The voltage from the power supply is increased from 0.1V to 5.0V which covers the voltage range of the most of the microprocessors [30], [31]. Fig. 20 demonstrates when the DoS attack is successful for various values of $V_{attack,L}$ where the attack indicator is 0 if the attack fails and 1 if the attack is successful. From Fig. 20, the DoS attack is successful if $V_{attack,L}$ is between 2.2V and 5.0V. When $V_{attack,L}$ is smaller than 3.5V, the current flows through the termination resistors which induces a voltage drop between CANH and CANL even under the DoS attack. However, as demonstrated in Fig. 21(a), V_{Diff} becomes less than the decision threshold for determining the dominant bit (i.e., 0.8V in this CAN transceiver.) when transmitting a dominant bit. If $V_{attack,L}$=5.0V, the voltages of CANH and CANL are the same as shown in Fig. 21(b), so the CAN bus represents a recessive bit all the time. Hence, the CAN bus is in the idle state regardless of transmitting dominant and recessive bits.



(a) Without the proposed IRS.

(b) With the proposed IRS.

Fig. 19: Message exchange through the CAN bus under the DoS attack. The DoS attack is launched from 11 to 31 seconds by ECU A. The message indicator at t_0 is 1 if the message is received at t_0 and 0 if a message is not received. (a) The messages are not exchanged during the attack. (b) The attack is mitigated by the proposed IRS since the message is exchanged during the attack.

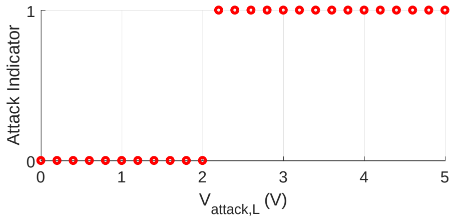
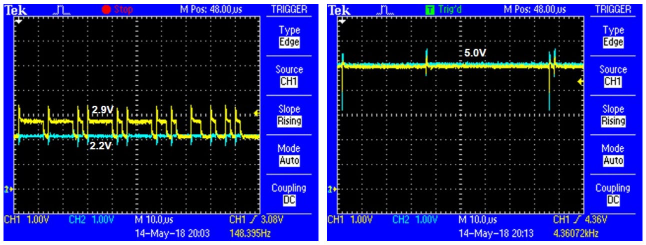


Fig. 20: The minimum value of $V_{attack,L}$ that leads to a successful DoS attack. The attack indicator is either 0 meaning a failure of the attack or 1 meaning a success of the attack. The DoS attack becomes successful from $V_{attack,L}$=2.2V.

D. Forced Retransmission Attack

We design an attack scenario which launches the forced retransmission attack using ECU A. ECU A initially sets P_H of its Arduino board to the input mode for 10 seconds and changes it to the high voltage output mode until t=30 second in order to launch the forced retransmission attack while keeping



(a) $V_{attack,L}$=2.2V

(b) $V_{attack,L}$=5.0V

Fig. 21: Voltages of the CAN bus for $V_{attack,L}$=2.2V and 5.0V at which the DoS attack is successful. (a) When $V_{attack,L}$=2.2V, V_{Diff} is less than the threshold for determining the dominant bit (0.8V). (b) When $V_{attack,L}$=5.0V, V_{Diff} is always 0.0V since both CANH and CANL are set to 5.0V.

P_L in the input mode. Then, ECU A sets P_H back to the input mode to stop the attack. The settings of the CAN bus speed, the periodic message exchange between ECUs B and C, and the oscilloscope are identical to the settings used in the DoS attack.

Fig. 22 demonstrates the message retransmission under the forced retransmission attack. Although ECU C transmits a message every 1 second, two consecutive messages are spaced by about 30μs, which indicates the message retransmission. The voltage of CANH could not be maintained at 5.0V since the Arduino board cannot generate the current large enough to let the voltage difference between CANH and CANL be greater than 3.5V. Fig. 23(a) shows the message exchange between ECUs B and C under the forced retransmission attack according to the attack scenario. The message is received by ECU B every 1 second before the attack occurs. However, when the attack occurs at t=10 second, the message is retransmitted, and thus the interarrival time between two consecutive messages is about 132μs. As the forced retransmission attack is stopped at t=30 second, the message is exchanged every 1 second as normal.

In order to demonstrate that the proposed IRS can mitigate the forced retransmission attack, we implement the proposed IRS at ECU A. Then, ECU A launches the attack between 10-30 seconds. Since the message is exchanged every 1 second during the attack, the proposed IRS successfully mitigates the attack as demonstrated in Fig. 23(b).

In order to determine the minimum voltage level that successfully launches the forced retransmission attack, we connect the power supply to CANH. Since the nominal voltage of the CAN bus in the idle state is 2.5V, we increase $V_{attack,H}$ from 2.5V to 5.0V using the power supply. The forced retransmission attack becomes successful from $V_{attack,H}$=4.5V, and Fig. 24 shows the voltages of the CAN bus at $V_{attack,H}$=5.0V. The magnitude of V_{Diff} satisfies the CAN protocol when transmitting both dominant and recessive bits as demonstrated Fig. 24. However, we observe that the duration of a recessive bit after a dominant is significantly smaller than the nominal bit duration (2μs) since the transition time from a dominant bit to a recessive bit increases.

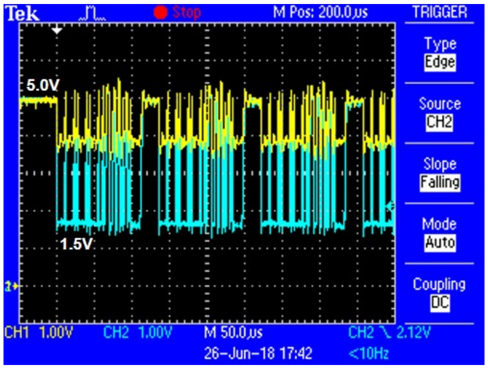
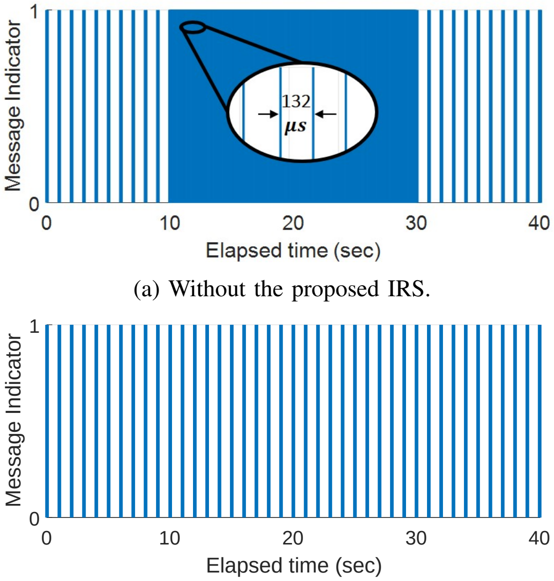


Fig. 22: Voltages of the CAN bus for $V_{attack,H}$=5.0V. The forced retransmission is successfully launched by ECU A. The same voltage waveform is repeated every 132μs which indicates the message retransmission.



(a) Without the proposed IRS.

(b) With the proposed IRS.

Fig. 23: Message exchange through the CAN bus under the forced retransmission attack. The attack is launched from 10 to 30 seconds. (a) The message is retransmitted during the attack. (b) The proposed IRS mitigates the attack since the message is transmitted every 1 second as normal.

Fig. 25 compares the bit length time τ_{bit} in the normal message transmission and under the forced retransmission attack. In the normal message transmission, the transition time is almost negligible as shown in Fig. 25(a), and τ_{bit} is 2μs. However, under the attack with $V_{attack,H}$=5.0V, it takes more than 1μs to change the voltage of CANL to represent a recessive bit after a dominant bit is transmitted. The duration of a recessive bit is no longer 2μs under the attack since the transition time increases. Hence, the duration of a recessive bit violates the CAN protocol. Since there are 7 transitions from a dominant bit to a recessive bit in the message with

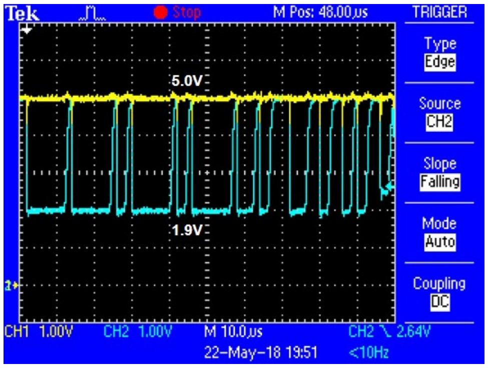
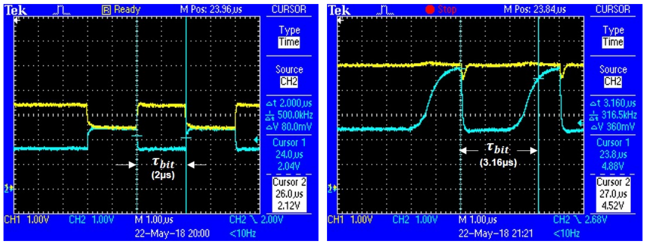


Fig. 24: Voltages of the CAN bus for $V_{attack,H}$ =5.0V at which the forced retransmission attack becomes successful using the power supply.



(a) Normal

(b) $V_{attack,H}$=5.0V

Fig. 25: Bit length time τ_{bit} in the normal message transmission and under the forced retransmission attack with $V_{attack,H}$=5.0V. (a) τ_{bit} is $2\mu s$ in the normal message transmission. (b) τ_{bit} is increased since the transition time of CANL from a dominant bit to a recessive bit increases under the attack.

ID 1 and data 1, we compute the average bit length time for various values of $V_{attack,H}$ from 2.5V to 5.0V as summarized in Table II.

TABLE II: Average bit length time for various values of $V_{attack,H}$ when the CAN bus speed is 500kbps.

$V_{attack,H}$	2.5V	3.0V	3.5V	4.0V	4.5V	5.0V
Average τ_{bit}	2.00μs	2.24μs	2.86μs	2.98μs	3.07μs	3.16μs

VIII. Conclusion

In this paper, we investigated the new attack surfaces if the extra wires given to the VIDS are maliciously used by the adversary. We proposed the three voltage-based attacks, namely, the overcurrent attack in which the microcontroller of the compromised ECU is damaged by overcurrent, the DoS attack in which all the message transmission through the CAN bus is blocked, and the forced retransmission attack in which a message is retransmitted by the targeted ECU. In order to defend against the proposed attacks, we proposed a hardware-based IRS that isolates the VIDS from the CAN bus immediately after the voltage-based attacks occur. We demonstrated the voltage-based attacks on the CAN bus testbed and showed that the proposed IRS can mitigate the voltage-based attacks. Our work suggests that the wires connecting the VIDS to the CAN bus may introduce the new attack surface to the CAN bus if the VIDS itself is compromised. Hence, in order to provide security assurance to the automobile, the defending mechanisms based on hardware have to be implemented together with the VIDS when attempting to leverage the voltage characteristics as a fingerprint of each ECU. As a future work, since the fuse has to be replaced manually after it blows out, we will investigate an IRS that has a simpler structure for isolating the compromised VIDS as well as recovers to the normal state automatically after the attack is removed.

Acknowledgment

This work was supported by ONR grant N00014-16-1-2710 P00002 and NSF grants CNS-1446866 and CNS-1544173.

References

[1] ISO, "International Standard ISO 11898-1 Road Vehicles-Controller Area Network (CAN), Part 1 Data Link Layer and Physical Signaling," 2015.

[2] ——, *International Standard ISO 17987 Road Vehicles-Local Interconnect Network (LIN), Part 1 General information and use case definition*, 2016.

[3] ——, *International Standard ISO 17458 Road Vehicles-FlexRay communication system, Part 1 General information and use case definition*, 2013.

[4] C. Miller and C. Valasek, "Remote exploitation of an unaltered passenger vehicle," in *Black Hat USA*, 2015.

[5] ——, "Adventures in automotive networks and control units," in *DEF CON21*, 2013.

[6] ——, "A survey of remote automotive attack surfaces," in *Black Hat USA*, 2014.

[7] K. Koscher *et al.*, "Experimental security analysis of a modern automobile," in *Proc. of the 2010 IEEE Symposium on Security and Privacy*. Washington, DC, USA: IEEE Computer Society, 2010, pp. 447–462.

[8] S. Checkoway *et al.*, "Comprehensive experimental analyses of automotive attack surfaces," in *Proceedings of the 20th USENIX Conference on Security*. Berkeley, CA, USA: USENIX Association, 2011, pp. 77–92.

[9] P. S. Murvay and B. Groza, "Source identification using signal characteristics in controller area networks," *IEEE Signal Processing Letters*, vol. 21, no. 4, pp. 395–399, April 2014.

[10] K.-T. Cho and K. G. Shin, "Fingerprinting electronic control units for vehicle intrusion detection," in *25th USENIX Security Symposium*. Austin, TX: USENIX Association, 2016, pp. 911–927.

[11] ——, "Error handling of in-vehicle networks makes them vulnerable," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '16. New York, NY, USA: ACM, 2016, pp. 1044–1055.

[12] T. Hoppe *et al.*, "Security threats to automotive CAN networks - practical examples and selected short-term countermeasures," in *Proceedings of the 27th International Conference on Computer Safety, Reliability, and Security*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 235–248.

[13] M. Müter and N. Asaj, "Entropy-based anomaly detection for in-vehicle networks," in *2011 IEEE Intelligent Vehicles Symposium (IV)*, June 2011, pp. 1110–1115.

[14] S. U. Sagong, X. Ying, A. Clark, L. Bushnell, and R. Poovendran, "Cloaking the clock: Emulating clock skew in controller area networks," in *Proceedings of the 9th ACM/IEEE International Conference on Cyber-Physical Systems*, ser. ICCPS '18. Piscataway, NJ, USA: IEEE Press, 2018, pp. 32–42. [Online]. Available: https://doi.org/10.1109/ICCPS.2018.00012

[15] K.-T. Cho and K. G. Shin, "Viden: Attacker identification on in-vehicle networks," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '17. New York, NY, USA: ACM, 2017, pp. 1109–1123. [Online]. Available: http://doi.acm.org/10.1145/3133956.3134001

[16] W. Choi, K. Joo, H. J. Jo, M. C. Park, and D. H. Lee, “Voltageids: Low-level communication characteristics for automotive intrusion detection system,” *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 8, pp. 2114–2129, Aug 2018.

[17] I. Foster, A. Prudhomme, K. Koscher, and S. Savage, “Fast and vulnerable: A story of telematic failures,” in *9th USENIX Workshop on Offensive Technologies (WOOT 15)*. Washington, D.C.: USENIX Association, 2015. [Online]. Available: https://www.usenix.org/conference/woot15/workshop-program/presentation/foster

[18] C. Carver, J. Hill, J. R. Surdu, and U. W. Pooch, “A methodology for using intelligent agents to provide automated intrusion response,” in *Proceedings of the IEEE Systems, Man, and Cybernetics Information Assurance and Security Workshop, West Point, NY*, 2000, pp. 110–116.

[19] Bosch, “CAN Specification Version 2.0,” 1991.

[20] Texas Instruments, “Introduction to the Controller Area Network (CAN),” 2016.

[21] O. Avatefipour and H. Malik, “State-of-the-art survey on in-vehicle network communication can-bus security and vulnerabilities,” in *International Journal of Computer Science and Network*, December 2017, pp. 720–727.

[22] Microchip, “MCP2551 CAN transceiver Datasheet,” 2016.

[23] NXP, “TJA1043 CAN Transceiver Datasheet,” 2017.

[24] Texas Instruments, “Overview of 3.3V CAN (Controller Area Network) Transceiver,” 2013.

[25] ——, “SN65HVD23x 3.3-V CAN Bus Transceivers,” 2015.

[26] ——, “Controller Area Network Physical Layer Requirements,” 2008.

[27] Arduino, “Arduino UNO Rev3 Technical Specification,” https://store.arduino.cc/usa/arduino-uno-rev3, accessed: 2018-05-07.

[28] Renesas, “V850/SA1 Application Note,” 2000.

[29] Texas Instruments, “Am335x Sitara Processors Datasheet,” 2016.

[30] Microchip, “ATmega 328P Automotive Datasheet,” 2015.

[31] NXP, “MPC561/MPC563 Reference Manual,” 2005.

[32] The European Union, “Directive 98/69/EC of the European Parliament and of the Council,” 1998.

[33] California Air Resource Board, “HD OBD Regulatory Documents,” 2012.

[34] Hyundai, “Hyundai global diagnostic system.” https://hyundai.service-solutions.com/en-US/Pages/ItemDetail.aspx?SKU=GDSM-LTKITH, June 2018, accessed: 2018-06-25.

[35] Ross Tech, “Volkswagen vag-com diagnostic system.” http://www.ross-tech.com/vag-com/index.html, June 2018, accessed: 2018-06-25.

[36] R. Rieke, M. Seidemann, E. K. Talla, D. Zelle, and B. Seeger, “Behavior analysis for safety and security in automotive systems,” in *2017 25th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*, March 2017, pp. 381–385.

[37] Atmel, “The Atmel-ICE Debugger User Guide,” 2016.

[38] Renesas, “V850E2/FF4-G,” 2014.

[39] J. Nilsson and S. Riedel, *Electric Circuits*, 10th ed. Pearson, 2014.

[40] LittelFuse, “Radial Lead Fuses 272/273/274/278/279 Series Very Fast-Acting Fuses Datasheet,” 2018.

[41] ——, “Fuse Characteristics, Terms and Condition Factors,” 2014.

[42] Y.-S. Chiu, K.-S. Chang, R. W. Johnstone, and M. Parameswaran, “Fuse-tethers in mems: theory and operation,” in *Canadian Conference on Electrical and Computer Engineering, 2005.*, May 2005, pp. 1517–1520.

[43] X. Dinga, W. Loub, and Y. Feng, “A controllable ic-compatible thin-film fuse realized using electro-explosion,” in *AIP Advances, 2016.*, vol. 6, 2016.