Physical Layer Data Manipulation Attacks on the CAN Bus

Abdullah Zubair Mohammed
Virginia Tech
abdullahzubair@vt.edu

Yanmao Man
University of Arizona
yman@email.arizona.edu

Ryan Gerdes
Virginia Tech
rgerdes@vt.edu

Ming Li
University of Arizona
lim@arizona.edu

Z. Berkay Celik
Purdue University
zcelik@purdue.edu

***Abstract*—The Controller Area Network (CAN) bus standard is the most common in-vehicle network that provides communication between Electronic Control Units (ECUs). CAN messages lack authentication and data integrity protection mechanisms and hence are vulnerable to attacks, such as impersonation and data injection, at the digital level. The physical layer of the bus allows for a one-way change of a given bit to accommodate prioritization; *viz.* a recessive bit (`1`) may be changed to a dominant one (`0`). In this paper, we propose a physical-layer data manipulation attack wherein multiple compromised ECUs collude to cause `0`→`1` (i.e., dominant to recessive) bit-flips, allowing for arbitrary bit-flips in transmitted messages. The attack is carried out by inducing transient voltages in the CAN bus that are heightened due to the parasitic reactance of the bus and non-ideal properties of the line drivers. Simulation results indicate that, with more than eight compromised ECUs, an attacker can induce a sufficient voltage drop to cause dominant bits to be flipped to recessive ones.**

I. Introduction

The Controller Area Network (CAN) bus is the most widely adapted in-vehicle communication network protocol in the automotive industry. The CAN bus provides communication between all the Electronic Contol Units (ECUs) of the vehicle. An ECU is an embedded system controlling a specific function of the vehicle such as engine control, transmission, entertainment units, etc. As many as 150 ECUs are installed in a modern vehicle to provide highly advanced safety and infotainment features [1]. Hence, a secure and reliable communication between the ECUs is of paramount importance, as the safety of the passengers directly depends on it.

The CAN communication protocol is broadcast in nature. Additionally, due to resource constraints, there is a lack of message authentication and data integrity mechanisms, which leads to a large attack surface. For example, an attacker may launch masquerade attacks where victim ECUs are impersonated by one or more attacker-compromised ECUs [2]. The attacker can reprogram those compromised ECUs in a local [3] or remote [4] manner. Furthermore, denial-of-service attacks can be launched by exploiting the error-handling schemes of the CAN protocol [5], [6].

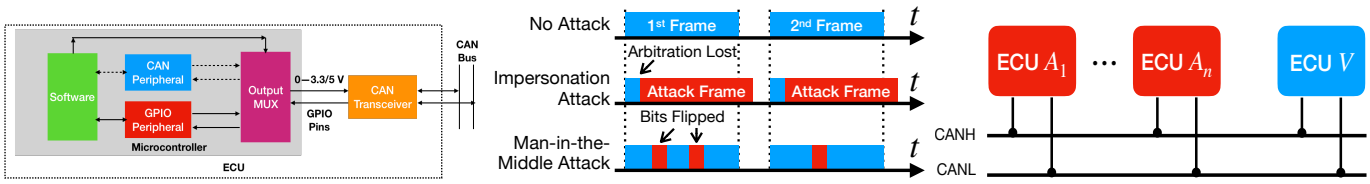
Since the ECUs have limited computation power, as well as small payloads of CAN frames, many state-of-the-art defenses against impersonation attacks adopt physical layer identification (PLI) methods. Without modifying the CAN protocol, these schemes authenticate each ECU by assigning a unique fingerprint based on their physical layer characteristics, such as timing [7] or voltage features [8]–[11]. These intrusion detection systems (IDS) compare the fingerprints of a sampled CAN frame with the fingerprints stored in a database to identify the ECUs, and a mismatch indicates an impersonation attack.

However, these voltage-based IDS have been shown to be vulnerable against recent, more advanced attacks [2], [12], in which the attacker can evade detection by gradually shifting the fingerprint of the impersonated ECU to the attacker-compromised ECU. They accomplish so by carefully choosing either the frequency of the injected frame, or the prefix length of the injected frames. Although these attacks are demonstrated to be feasible, they are difficult to launch because the attacker has to strictly follow the CAN protocol specified by the CAN peripheral on each ECU's microcontroller.

In this paper we propose a new way to launch arbitrary bit-flipping attacks on the CAN bus; i.e., the attacker aims to alter any bit within an ongoing CAN frame. There are two main challenges. First, the attacker needs to bypass typical CAN transmissions mechanisms (i.e., the CAN peripheral of the ECU) to be able target individual bits. Secondly, to change a recessive bit to a dominant one requires a signal cancellation mechanism inherent to an ECU.

We explain how an attacker can indeed bypass the CAN peripheral and leverage the GPIO peripheral, common to almost all microcontrollers used in ECUs, to allow for arbitrary bit injection. With this capability, the Dolev-Yao model [13] is applicable to CAN bus attacks for the first time. Not only does our threat model make existing ECU impersonation attacks easier to launch, we take a step further and propose a Man-in-the-Middle (MitM) attack against CAN, in which a transmitted frame can be modified on-the-fly. To do this, the attacker exploits the non-ideal characteristics of the CAN bus to manipulate the data in the form of bit-flips. The reactance of the CAN bus (a twisted pair transmission line) and the parasitic capacitance of the transistors in the CAN transceiver can be leveraged to induce transients that cause a transition from a dominant (`LOW`) to recessive (`HIGH`) bit. Specifically, in our attack multiple compromised ECUs perform high speed switching to generate transients that are strong enough to *flip* a bit that is being transmitted on the CAN bus. Our simulation results, using a practical SPICE model of the CAN transceiver and bus, show that eight or more synchronized, attacker-controlled ECUs can cause the bit-flips in either direction.

Workshop on Automotive and Autonomous Vehicle Security (AutoSec) 2022
24 April 2022, San Diego, CA, USA
ISBN 1-891562-75-4
https://dx.doi.org/10.14722/autosec.2022.23047
www.ndss-symposium.org



(a) (b) (c)

Fig. 1: (a) Architecture of an Electronic Control Unit (ECU); (b) Impersonation attacks versus man-in-the-middle attacks. When there is no attack, two legitimate frames are transmitted as normal. Under an impersonation attack, the attack ECUs first need to win the arbitration over from the legitimate ECU and then continue to broadcast an attack frame with the legitimate ECU's ID in it. In our attack, the adversary tampers the messages on the fly, flipping arbitrary bits; (c) Multiple compromised ECUs (A_1 to A_n) collaboratively alter the messages sent by ECU V.

II. Threat Model and Background

In this section, the aim and capabilities of the adversary are outlined and the technical background necessary to understand the theory of attack is discussed.

A. Physical Layer of CAN protocol

At the physical layer, CAN bus has two signal lines, CAN high (CANH) and CAN low (CANL), terminated by two 120 Ω resistors. The CAN protocol employs differential signaling to represent a bit. During the transmission of the bit-1 (recessive), the voltage on CANH and CANL is equal and therefore differential voltage is 0 V. When transmitting bit-0 (dominant), CANH and CANL are set to HIGH and LOW resulting in a differential voltage of around 2 V, depending on the supply voltage V_{cc}.

The architecture of an ECU is shown in Figure 1a. The CAN transceiver acts as an interface between the microcontroller and the CAN bus. It translates the single-ended voltage signal of the microcontroller to the differential voltage on the CAN bus.

B. Threat Model

We consider an adversary with a goal of flipping one or more bits of a CAN frame that is being transmitted. The flipped bits can be in the ID field, or the overhead, thus breaking the integrity and authenticity, which differs from existing impersonation attacks that can only compromise the authenticity (Figure 1b).

To do that, we assume that the adversary can compromise multiple ECUs (Figure 1c), where they reprogram each ECU in order to bypass their CAN peripheral (Figure 1a). The adversary can achieve this by using Return Oriented Programming [14] method to execute the functions required to bypass CAN peripheral and map the microcontroller pins to a GPIO peripheral. In many automotive grade microcontrollers, such as Texas Instruments TMS470M [15] and TMS570 [16], the pins that are connected to the CAN transceiver are reconfigurable and can be used both for CAN functionality and for general purpose input output (GPIO) functions and it only needs programming of one register to change the functionality. Moreover, the high clock rate (8 MHz) of these microcontrollers enable the adversary to easily switch at least 10 times faster than the standard CAN data rate.

Once an ECUs is compromised, the adversary can use its GPIO peripheral (instead of CAN peripheral) to transmit bits

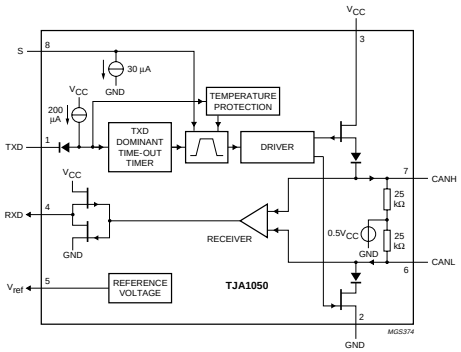


Fig. 2: Schematic of TJA 1050 CAN transceiver [17]. The FETs at the output are switched ON and OFF by the driver when the dominant and recessive bits are transmitted, respectively.

to the CAN transceiver at the data rate of their choice and also override the arbitration process of the CAN protocol. This way, the rogue ECU, an ECU controlled by the adversary, can transmit a dominant bit even when a legitimate ECU is already transmitting the dominant bit on the bus. The CAN transceiver connected to the rogue ECU is not under the adversary's control, and therefore the output voltage on the bus will still take the nominal values of the CAN bus. The adversary has information of data rates supported by the transceiver and has the ability to design the attack signal for all standard CAN data rates. All the rogue ECUs are synchronized with each other.

C. Theory of Attack

Before discussing the theory of attack, we go over the working of a CAN transceiver. The schematic of NXP TJA1050 transceiver [17] is shown in Figure 2. The voltage on the lines CANH and CANL depends on the state of the two field-effect transistors (FETs), the p-channel FET, connected to the CANH line and the n-channel FET connected to CANH. When the ECU is transmitting the dominant bit, the FETs are switched on by the driver causing the voltage to drop across them to develop a differential voltage on the CAN bus. During this stage, p-channel and n-channel FETs actively pull the CANH and CANL lines to V_{cc} and ground respectively. Whereas, in the recessive bit transmission, the FETs are off and are in the high-impedance (high-Z) state. As no current is flowing, no voltage difference develops between the CANH and CANL lines. And when another ECU is transmitting the dominant bit, the high-Z lines of the recessive ECU follow the output voltage of the dominant ECU.

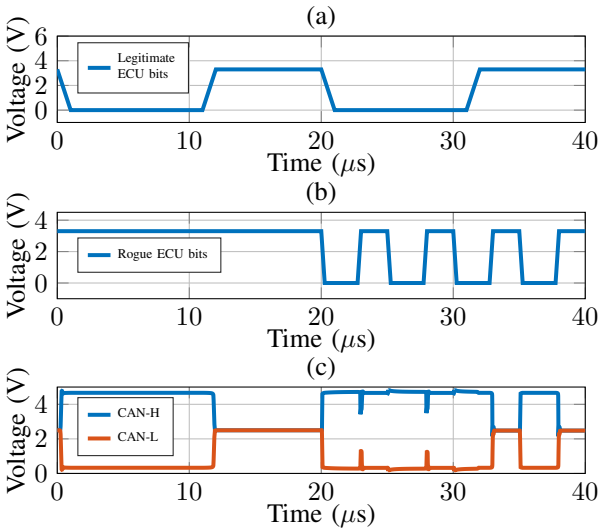


Fig. 3: Theory of Attack: (a) The bit-pattern of the legitimate ECU; (b) The bit-pattern of the rogue ECU; (c) Voltage on the CANH and CANL lines. The attack starts at time, t = 20 µs.

We demonstrate the basic principal of the proposed attack in Figure 3. Figure 3a shows the input to the CAN transceiver of legitimate ECU from it's CAN peripheral and corresponds to the bit-pattern being transmitted (at the nominal data-rate of 100 kbps) by a legitimate ECU. Figure 3b shows the input to a rogue ECU's transceiver from the GPIO peripheral, switching at a higher data-rate of 400 kbps. It should be noted that 0 V (low) corresponds to dominant bit and 3.3 V (high) corresponds to the recessive bit. Figure 3c shows the voltages on the lines CANH and CANL. During the initial 20 µs, the bus is under no attack and we see nominal voltages on the lines. During the attack between 20 µs and 40 µs, the rogue ECU starts switching and we see discrepancies on the bus.

Legitimate ECU is transmitting dominant bit: In the Figure 3c, we observe transients on the CANH and CANL lines when the rogue ECU switches from dominant to recessive. The transients occur because, as discussed above, during the dominant transmission, the FETs in the transceiver of the legitimate ECU are actively pulling the CANH and CANL voltages to stay high and low respectively. Therefore, the FETs instantaneously pull the line voltages back when the rogue ECU switches from dominant to recessive, causing a brief voltage drop in the form of a transient. We seek to exploit these transients, by increasing the number of rogue ECUs and increasing their data-rate, to cause *sufficient* voltage drop to cause a *bit-flip* from dominant to recessive.

Legitimate ECU is transmitting recessive bit: In this state, the bus follows the state of rogue ECU because the FETs of legitimate ECU are in the high-Z state, as discussed above, and are not pulling the line voltages during recessive transmission. Therefore, the bus goes dominant when the rogue ECU goes dominant, causing a bit-flip.

We demonstrate the theory of attack experimentally with one legitimate ECU and one rogue ECU that switches between dominant and recessive bits at 1 Mbps. Due to rapid switching of rogue ECU, the voltage drops and transients that appear on the bus when legitimate ECU is transmitting dominant bit can be seen in the oscilloscope capture in Figure 4.

D. Attack Impact

Since our attack compromises both the authenticity and integrity of CAN messages, it undermines most existing PLI-based CAN bus intrusion detection systems that require retraining [8]–[11]. To evade their detection, the basic idea is to carefully choose the number of flipped bits at each step in order to gradually shift the fingerprinting profile of victim ECU, i.e., a hill-climbing-style attack on the bit level [12].

Moreover, our threat model makes existing inter- and intra-frame CAN bus attacks [2], [12] easier too because the attacker can bypass the CAN peripheral and gain access to the GPIO pins directly; hence we do not need to strictly follow the CAN protocol. For example, the DUET attack [2] has to involve an accomplice ECU for synchronization; with our threat model, an attacker can passively listen for the correct message ID and manipulate the bus by itself at will. Similarly, RAID [2] relies on randomized IDs at retraining to defeat DUET, but the attacker ECU in our threat model can read the state of the GPIO pins at a rate higher than the transmission rate, which allows them to manipulate the bus despite randomization.

III. Attack Design and Simulation Setup

The objective of the attacker is to flip the bit that is being transmitted on the bus. It is trivial to cause the flip from recessive to dominant, and the adversary needs only one rogue ECU to transmit a dominant bit. This is because, according to CAN protocol, in the case of a collision, the dominant bits win, and hence the legitimate ECU stops transmitting.

For the dominant to recessive flip, the adversary needs to bring down the differential voltage of the bus to lower than the detection threshold, around 0.7 V [17], and hold it for a time that the receiver takes to sample the bus voltage. To achieve this, the adversary needs to compromise multiple ECUs and have them transmit at the highest data rate so that the transient on the CAN bus is registered as a voltage drop.

To observe the effects of the transmission line and the parasitic capacitance of the FETs in the transceiver, we modeled the adversarial setup on SPICE.

A. SPICE circuit

The CAN transceivers are modeled according to the schematic in Figure 2 using practical p-channel and n-channel

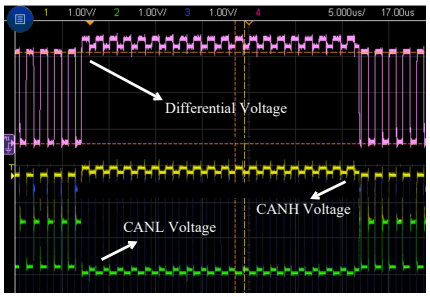
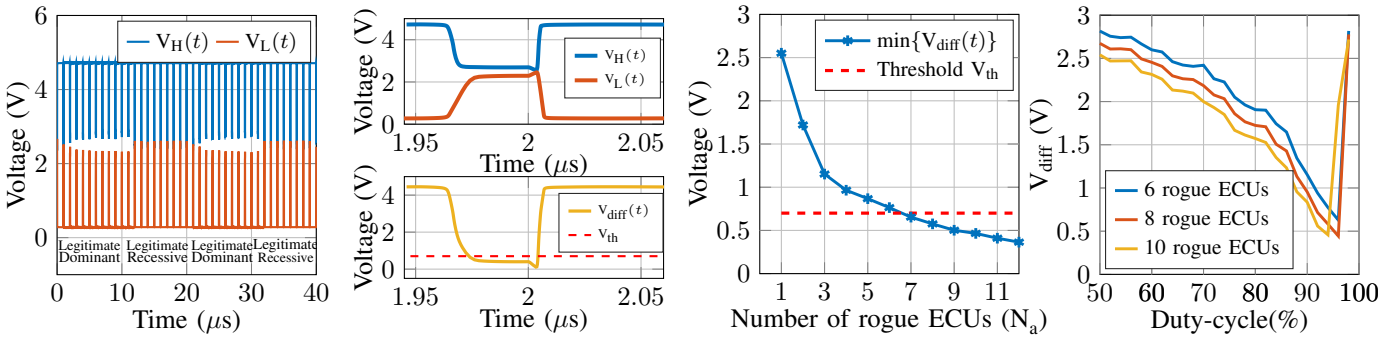


Fig. 4: Illustration of the theory of attack: The oscilloscope capture shows the CANH, CANL and the differential (CANH - CANL) voltages when a compromised ECU is switching rapidly.



(a) (b) (c) (d)

Fig. 5: Simulation Results: (a) The voltage on CANH and CANL lines for 10 rogue ECUs and a duty-cycle of 94%; (b)-(top) The zoomed-in transients on CANH and CANL lines; (b)-(bottom) The zoomed in differential voltage during the transient. (c) Minimum differential voltage observed on the bus vs. Number of rogue ECUs; (d) Minimum differential voltage observed on the bus vs. duty-cycle of rogue ECUs

MOSFETs [18]. The CAN bus is modeled as a transmission line with the transmission line parameters of a 120 Ω unshielded twisted pair (UTP) cable (SAE J1939 standard) [19]. The sub-circuit model of the transmitter part of the CAN transceiver and the adversarial model with two rogue ECUs is shown in Figures 6a and 6b in the Appendix. The pulse voltage sources act as the driver and control the data rate of the transmission. The data rate of the legitimate ECU is set to 100 kbps while the rogue ECUs transmit at 1 Mbps.

By feeding the output wave-forms of the SPICE circuit to an optimizer, we converged on the attack parameters (number of rogue ECUs, their data pattern) required to achieve the voltage drop sufficient for a dominant to recessive bit-flip. The definition of the optimizer used is given in the Appendix under Section A. The optimizer converged to the following attack parameters: Number of rogue ECUs, N_a = 10 and time ON (transmitting dominant bit) for rogue ECUs, t_a = 940 ns (duty-cycle (dc) of 94%). It can be understood that the higher the number of rogue ECUs, the more difficult it will be for the legitimate ECU to pull back the line voltages to the dominant voltage values during the transient.

IV. Simulation Results

The simulation parameters are as follows: the legitimate transmitter ECU transmits alternate dominant and recessive bits at 100 kbps while the rogue ECUs transmit a periodic rectangular pulse at 1 Mbps. In each period, the rogue ECUs stays dominant for 940 ns and switches to recessive transmission for 60 ns (i.e., 94% duty-cycle). A transmission line of length 10 m is considered between the legitimate transmitter ECU and legitimate receiver (victim) ECU. A transmission line of 1 m is considered between the victim ECU and each rogue ECU.

The line voltages for two cycles of legitimate ECUs dominant and recessive transmission are shown in Figure 5a. High transients can be observed on CANH and CANL lines during the legitimate ECU's dominant transmission. The zoomed-in line voltages and the differential voltage ($V_{diff}(t) = V_H(t) - V_L(t)$) during one transient is displayed in Figure 5b. It can be seen that, during transient, the differential voltage drops below the detection threshold (V_{th}) and thus will cause a bit-flip from dominant to recessive. The bit-flip from recessive to dominant is, as discussed earlier, trivial. This can be observed in Figure 5b when the legitimate ECU is transmitting recessive bit, the line switches to dominant when the rogue ECUs are dominant.

The performance of the adversary with an increasing number of rogue ECUs is shown in Figure 5c. As expected, the minimum differential voltage observed on the bus decreases with an increase in the number of rogue ECUs. At N_a = 7, the minimum value of V_{diff} is sufficient to cause a bit-flip. The variation of the differential voltage with respect to the duty-cycle of the rogue ECUs is shown in Figure 5d. The minimum differential voltage decreases as the duty-cycle increases and abruptly increases at 98%. This is possible because at 98% duty-cycle, the rogue ECUs switch to recessive for only 20 ns and such a fast transition may not be sampled by the transceiver, resulting in no effect of switching.

V. Ongoing Work and Conclusions

The hardware realization of the proposed attack is in progress. We are also working on bringing down the number of rogue ECUs that need to be compromised by the adversary for a successful bit-flip attack. Note that 8 compromised ECUs out of around 150 is already a small proportion, but the proposed attack will be even stronger and more feasible with fewer compromised ECUs. We plan to achieve this by studying the performance for additional attack parameters, such as the variable switching speed of rogue ECUs, variable delay among the rogue ECUs, and varying transmission line lengths between the ECUs. We will also be improving the simulation results by using models of CAN transceivers from other vendors, in addition to TJA 1050, on which we currently base our model.

With our current simulation results, we demonstrate that 8 or more rogue ECUs can, at will, cause the bits on the CAN bus to flip from dominant to recessive and vice-versa. This will enable the adversary to manipulate the data being exchanged between legitimate ECUs leading to severe consequences in the vehicle's operation.

Acknowledgements: We wish to thank Mahsa Foruhandeh (Virginia Tech) for assistance in the experiments with CAN transceivers. This work is supported in part by the National Science Foundation (Grant CNS-1801611 and CNS-1658225).

REFERENCES

[1] "How software is eating the car," https://spectrum.ieee.org/software-eating-car, [Online; accessed 11-January-2022].

[2] R. Bhatia, V. Kumar, K. Serag, Z. B. Celik, M. Payer, and D. Xu, "Evading voltage-based intrusion detection on automotive can," in *Network and Distributed System Security Symposium (NDSS)*, 2021.

[3] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham *et al.*, "Experimental security analysis of a modern automobile," in *IEEE symposium on security and privacy (IEEE S&P)*, 2010, pp. 447–462.

[4] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, T. Kohno *et al.*, "Comprehensive experimental analyses of automotive attack surfaces." in *USENIX Security Symposium*, vol. 4, no. 447-462. San Francisco, 2011, p. 2021.

[5] K.-T. Cho and K. G. Shin, "Error handling of in-vehicle networks makes them vulnerable," in *Conference on Computer and Communications Security (CCS)*, 2016, pp. 1044–1055.

[6] K. Serag, R. Bhatia, V. Kumar, Z. B. Celik, and D. Xu, "Exposing new vulnerabilities of error handling mechanism in CAN," in *USENIX Security Symposium*, 2021, pp. 4241–4258.

[7] K.-T. Cho and K. G. Shin, "Fingerprinting electronic control units for vehicle intrusion detection," in {*USENIX*} *Security Symposium*, 2016, pp. 911–927.

[8] M. Kneib and C. Huth, "Scission: Signal characteristic-based sender identification and intrusion detection in automotive networks," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018, pp. 787–800.

[9] 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*, 2017, pp. 1109–1123.

[10] M. Kneib, O. Schell, and C. Huth, "Easi: Edge-based sender identification on resource-constrained platforms for automotive networks." in *NDSS*, 2020.

[11] 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, 2018.

[12] M. Foruhandeh, Y. Man, R. Gerdes, M. Li, and T. Chantem, "Simple: Single-frame based physical layer identification for intrusion detection and prevention on in-vehicle networks," in *Proceedings of the 35th Annual Computer Security Applications Conference*, 2019, pp. 229–244.

[13] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on information theory*, vol. 29, no. 2, pp. 198–208, 1983.

[14] N. R. Weidler, D. Brown, S. A. Mitchell, J. Anderson, J. R. Williams, A. Costley, C. Kunz, C. Wilkinson, R. Wehbe, and R. Gerdes, "Return-oriented programming on a resource constrained device," *Sustainable Computing: Informatics and Systems*, vol. 22, pp. 244–256, 2019.

[15] *TMS470M Series Technical Reference Manual (TRM)*, Texas Instruments, 2013. [Online]. Available: https://www.ti.com/lit/ug/spnu495c/spnu495c.pdf

[16] *TMS570LS 16,32-Bit RISC Flash Microcontroller Technical Reference Manual*, Texas Instruments, 2018. [Online]. Available: https://www.ti.com/lit/ug/spnu517c/spnu517c.pdf

[17] *TJA1050-High speed CAN transceiver*, NXP, 2002. [Online]. Available: https://www.nxp.com/docs/en/data-sheet/TJA1050.pdf

[18] *EM6M2 datasheet*, Rohm, 2016. [Online]. Available: https://fscdn.rohm.com/en/products/databook/datasheet/discrete/transistor/mosfet/em6m2t2r-e.pdf

[19] T. K. Truong, "Twisted-pair transmission-line distributed parameters," *EDN Mag*, 2000.

APPENDIX

A. Optimization Definition

A genetic algorithm optimizer is used to find a solution for the adversarial model to achieve minimum differential voltage on the bus when the legitimate ECU is transmitting dominant bit. The variables to the optimizer are the number of rogue ECUs (N_a) and the time for which rogue ECUs are in dominant state (t_a). The optimization problem for our adversarial setup is defined as follows,

$$\begin{aligned} \underset{t_a, N_a}{\text{minimise}}\ & \min(V_H(t, t_a, N_a) - V_L(t, t_a, N_a)) \\ \text{s.t } 0 \leq & t_a \leq 1000 \text{ ns}, \\ & 0 < N_a \leq 10, \end{aligned} \tag{1}$$

where

$V_H(t, t_a, N_a)$ is the voltage on the CAN high line at instantaneous time, t, $0 \leq t \leq t_h$.

$V_L(t, t_a, N_a)$ is the voltage on the CAN low line at instantaneous time, t, $0 \leq t \leq t_h$.
Note: The voltages V_H and V_L also depend on the number of rogue ECUs trying to manipulate the data.

t_h is the time ON for legitimate ECU (transmitting dominant bit),

B. SPICE Circuits

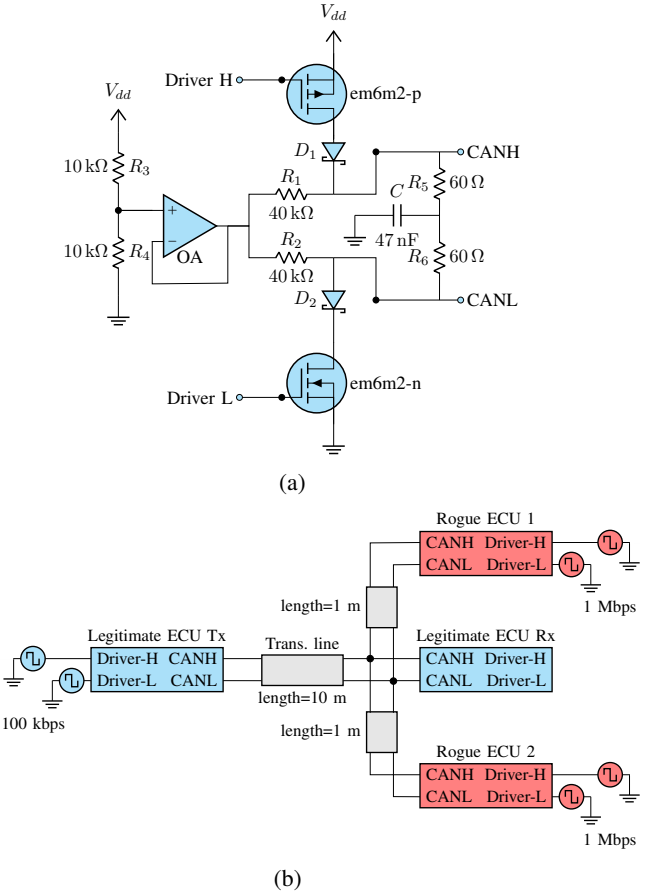


Fig. 6: (a) The sub-circuit model of CAN transmitter (b) The adversarial model with 2 rogue ECUs attacking the bus.