A Model-Based Method for Enabling Source Mapping and Intrusion Detection on Proprietary Can Bus

Jia Zhou, Guoqi Xie, *Senior Member, IEEE*, Haibo Zeng, *Member, IEEE*, Weizhe Zhang, *Senior Member, IEEE*, Laurence T. Yang, *Fellow, IEEE*, Mamoun Alazab, *Senior Member, IEEE*, and Renfa Li, *Senior Member, IEEE*

***Abstract*— With the deep integration of the Internet of Things (IoT) technology and the increase of computational power and memory, vehicles can also serve as the infrastructures for Intelligent Transportation System (ITS), e.g., as fog nodes. However, when connecting vehicles to the internet, alongside with the benefits it brings, it also opens many new challenges such as security attacks. Controller Area Network (CAN) is one of the main in-vehicle communication protocols in modern cars. Its lack of sender verification mechanism makes CAN particularly vulnerable to cyber-attacks including masquerade attack. Fingerprinting Electronic Control Units (ECUs) based on hardware characteristics has been proved feasible and effective on defending CAN buses. However, most state-of-the-art works exploited the supervised learning algorithm to identify the transmitter based on the signal characteristics. This makes the decision process hard to understand, and it also limits the deployment on proprietary CAN bus without prior knowledge. To solve this, we design a novel clock-skew-based approach capable of pinpointing the sender and detecting intrusion on proprietary CAN bus. We take a single CAN frame as the object for measurement, and adjust the measuring process such that our approach can be independent of the transmission time of frames. Based on the statistical analysis of data from real vehicles, we propose a novel box-plot algorithm based on score mechanism to filter the raw data. Finally, the clock skews are estimated and accumulated to build a linear model for representing the transmitter ECU. The evaluation results on one CAN prototype and two production vehicles show that our approach is able to well identify and differentiate ECUs on the bus without prior knowledge. The data processed by the proposed box-plot algorithm can describe the hardware characteristics of ECUs precisely. We also show the ability of our approach to protecting the CAN bus against the masquerade attack.**

***Index Terms*— Automotive security, controller area network, source mapping, intrusion detection.**

I. Introduction

BENEFITING from the Internet of Things (IoT) which highly integrates the advanced sensing, computing and communicating technologies, the Intelligent Transportation Systems (ITS) can not only connect traditional physical entities such as vehicles, roadside units and people, but also virtual world like social systems to enable a higher level of efficiency, safety and productivity [1]. To provide IoT services with high efficiency and low latency, the connected vehicles as well as the smart phones can also serve as the infrastructures for ITS in addition to the roadside base stations and intelligent traffic lights. For example, the fog-cloud computing employs the vehicles as the fog infrastructures to implement complex neural network models such as distributed deep learning (DDL) [2], [3], which can favour the deployment of resources-intensive ITS applications.

However, besides the benefits, the increasing connectivity exposes the internal communication systems of the related elements in the ITS to devastating cyber-threats [4]. The Controller Area Network (CAN) is wildly used on automotive network as well as the small-sized automation systems for enabling interconnection and interoperability among heterogeneous devices. Since there is no built-in security mechanism for access control or sender verification, the attacker can transmit arbitrary messages to manipulate the functions of automotive applications once it gains access to a target CAN bus [5].

As one promising solution, the device fingerprint which can provide integrity as well as non-repudiation is well studied to

Manuscript received 14 November 2021; revised 24 January 2022; accepted 4 February 2022. Date of publication 3 March 2022; date of current version 1 November 2023. This work was supported in part by the Peng Cheng Laboratory Project under Grant PCL2021A02, in part by the Key-Area Research and Development Program of Guangdong Province under Grant 2020B0101360001, and in part by the National Natural Science Foundation of China under Grant 61932010 and Grant 62002147. The Associate Editor for this article was H. Lv. *(Corresponding authors: Weizhe Zhang; Guoqi Xie; Laurence T. Yang; Mamoun Alazab.)*

Jia Zhou is with the Department of New Networks, Peng Cheng Laboratory, Shenzhen 518055, China (e-mail: zhouj02@pcl.ac.cn).

Guoqi Xie and Renfa Li are with the Key Laboratory for Embedded and Network Computing of Hunan Province, College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China (e-mail: xgqman@hnu.edu.cn; lirenfa@hnu.edu.cn).

Haibo Zeng is with the Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA 24060 USA (e-mail: hbzeng@vt.edu).

Weizhe Zhang is with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China, and also with the Department of New Networks, Peng Cheng Laboratory, Shenzhen 518055, China (e-mail: wzzhang@hit.edu.cn).

Laurence T. Yang is with the Department of Computer Science, St. Francis Xavier University, Antigonish, NS B2G 2W5, Canada (e-mail: ltyang@ieee.org).

Mamoun Alazab is with the College of Engineering, IT and Environment, Charles Darwin University, Casuarina, NT 0810, Australia (e-mail: alazab.m@ieee.org).

Digital Object Identifier 10.1109/TITS.2022.3153718

harden the security of CAN bus [6]. The device fingerprint can be generated from the signal characteristics on the bus medium to check whether the received data frame is falsified. The signal characteristics can be unique while remaining stable within boundaries for several months [7]. One way of constructing the fingerprint is based on the supervised learning algorithms. The general process of these approaches [8]–[12] can be separated as three phases. The first phase is to obtain the characteristics of physical signal from the perspective of voltage or time. Subsequently, statistical features in time and/or frequency domain are extracted from the measurements. The combination of these selected features is regarded as the device fingerprint for sending ECU. In the last phase, these values are fed into a supervised learning classifier to establish the relationships between measurements and ECUs. However, since the fingerprint generation process cannot be well explained, the output cannot be fully understood. Another limitation of these approaches is they cannot be applied on proprietary CAN bus without prior information since the labeled data is required for classifier generation.

The second way for generating the fingerprint exploits an physical model to explicitly describe the relationship between measurements and the hardware characteristics. Studies [13], [14] build the model to source the sending ECUs and defend against attack by utilizing the clock skew. Clock skew is introduced by the imperfect production process of the clock circuit, and it can be unique and constant, making it suitable for fingerprinting devices. However, their common downside is that the clock skew estimation depends on the measurements of arrival time from periodic traffic. They can not be well applied for aperiodic or sporadic frames. Furthermore, the irregularity for some periodic frames in data collected from real vehicles can result in the reduction in performance [14].

To mitigate these issues, we design a novel clock-skew-based approach to enable pinpointing the sender and detecting anomaly on proprietary CAN bus. In our work, the clock skew is calculated from the measurements of the frame-level signal. The cumulative sum of the computed clock skews is then used to build a linear model for representing the sender. Our main contributions are as follows.

1) We propose a model-based method which utilizes the skew in clocks to describe the relationship between the signal characteristics and the sending ECU. The decision process of our system can be clear and explainable based on the constructed model.
2) We compute the clock skew from the signal length of one single frame instance instead of time differences between consecutive periodic frames. Thus, our approach can be independent from arrival or transmission time of CAN frames. Our approach can be applied to any CAN frames regardless of their transmission periodicity (periodic, aperiodic, or sporadic).
3) We propose a novel box-plot algorithm for removing abnormal measurements based on the statistical analysis of data from real vehicles. The data filtered by the new box-plot algorithm can precisely describe the hardware characteristics of the sending ECU.
4) We evaluate our approach on a CAN bus prototype and two production vehicles. It is demonstrated that our approach can build the model for ECUs without any prior knowledge on the monitored CAN bus. ECUs on both prototype and real vehicles can be distinguishable, and the intrusion can be detected effectively.

II. Background

A. Primer on Can

1) Automotive CAN Bus: Today, CAN is the most popular communication protocol for automotive network and implemented on every production vehicle [10]. The safety-critical information such as engine and cruise control is exchanged on the CAN bus. In the automotive industry, the CAN bus is proprietary that the semantic knowledge and its sender of the transmitted CAN message cannot be obtained publicly. Due to its critical significance on controlling vehicles behaviours, the automotive CAN bus is becoming the target of cyber-attack.

2) Broadcast Nature and Message-Oriented Protocol: CAN is a multi-master and broadcast communications system. Every node can publish messages to the entire network and it can be received by all the rest nodes. There is no any field in CAN frames specified for its transmitter or receiver. The decision to drop or further process for one newly received frame is only made by checking the identifier.

3) CAN Data Transmission: The data in CAN is exchanged via the unit called data frame. The structure can be divided into five fields, including arbitration field, control field, data field, CRC (Cyclic Redundancy Check) field and ACK (Acknowledgement) field (can be seen in Figure 2). The arbitration field bears the identifier which can be used for identifying different frames as well as competing the rights of transmitting on the bus. For a given CAN network under normal circumstances, it is required that the identifier should be unique and can only be sent by one and only one ECU [15]. In our work, the messages with same identifier are considered as the same frame.

The electrical signal is transmitted on two wires called CAN high and CAN low respectively. They are pulled in the opposite directions to generate one differential signal. The bit 1 (called recessive bit) is transmitted when CAN high = CAN low, while the bit 0 (called dominant bit) is transmitted when CAN high > CAN low.

4) Arbitration and Acknowledgment: The mechanism of arbitration is needed when multiple nodes intend to publish data simultaneously. CAN adopts the process called bit-wise arbitration to determine which node can have the right to send data. The frame with lower identifier can win the arbitration and have the access to the bus exclusively. The winner keeps sending data until the ACK field. Then, all nodes other than the transmitter that have verified the correctness of the frame would pull up the differential signal to acknowledge for no errors during the ACK field.

B. Cause of Timing Difference in Signal

Here we discuss two hardware factors which can affect the duration for a given CAN signal.

1) Clock: The difference in clocks has been proved effective for triggering intrusion or sourcing the sender on automotive CAN bus in previous work [12]–[14]. It is the intrinsic physical characteristics in electronic devices. Thus, the electrical signal driven by the device can inherit the variations of characteristics from the clock. The related concepts of clock are reviewed next. There are two clocks denoted by $\mathbb{C}_1$ and $\mathbb{C}_2$.

- ***Clock Offset:*** The time difference between clocks, i.e., $\mathbb{C}_1(t) - \mathbb{C}_2(t)$ is the clock offset between two example clocks.
- ***Clock Frequency:*** It indicates how fast the clock runs, which is denoted by $\mathbb{C}'_1(t)$ for $\mathbb{C}_1$ at time t. It can be expressed as: $\mathbb{C}'_1(t) = \mathrm{d}C_1(t)/\mathrm{d}t$.
- ***Clock Skew:*** The difference in frequency between clocks, i.e., $\mathbb{C}'_1(t) - \mathbb{C}'_2(t)$ is the skew of these two example clock $\mathbb{C}_1$ and $\mathbb{C}_2$ at time t.

2) CAN Transceiver: Besides the clock, the CAN transceiver can affect the length of the signal. To communicate with nodes on the CAN bus, a CAN controller and a transceiver should be equipped (e.g., the ECU-k shown in Figure 1). The CAN controller is used for implementing the functions specified in the protocol, while the transceiver (which means transmit and receive) controls how the data of logical level is converted to the electrical representation transmitted on the bus wires [15]. Hence, the characteristics and noise such as jitter [16] can be expressed in the signal as well.

III. Related Work

The signal characteristics have proved to be unique for representing different ECUs and remain stable for several months [7], thus can be exploited for identifying the sending ECUs and reporting the intrusion. We divide these approaches into two categories, which are supervised learning algorithms based approaches and model based approaches, according to whether it can work without labeled data from the bus.

A. Supervised Learning Algorithms Based Approaches

The problem of sender identification can be regarded as the problem of classification. The frames that are coming from the same ECU are regarded as one class. The approaches in this category use the supervised learning algorithms to solve the classification problem.

Choi *et al.* [8] proposes an approach by using the voltage level measured from an pre-allocated bit string. All ECUs on the bus are required to be reprogrammed. Thus it reduces the the likelihood for deployment on real industrial network. Studies [9], [10] improve the method of extracting the signal characteristics using voltage. The voltage level of dominant bits of the recorded signal are divided into three individual sets to derive more information which can better describe the characteristics of the transmitter. EASI [11] proposes low-cost solutions which can favor the deployment on the automotive CAN bus. Other than voltage, BTMintor [12] measures the small but recognizable deviations of bit time to identify the differences in clocks of ECUs.

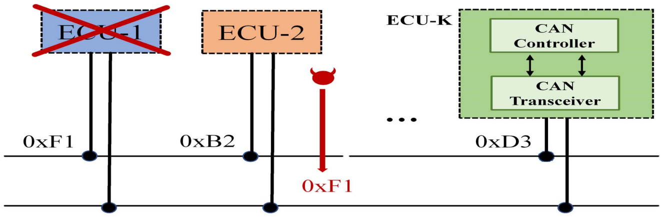


Fig. 1. Adversary model.

After collecting the signal characteristics, the above approaches extract the statistical features from the measurements as device fingerprint. The generated fingerprint is fed into an classifier for intrusion detection. The supervised learning algorithms are used to generate the classifier based on the labeled data. One common downside of these approaches is that they cannot be applied to an proprietary CAN bus without prior knowledge. Besides, the decision made by the classification model cannot be fully understood.

B. Model Based Approaches

The approaches in this category build the model to describe the relationship between signal characteristics and the sending device which can enable the decision process more clear and explainable.

Viden [17] uses the measurements of voltage to construct the model. The voltages of CAN high and CAN low when transmitting dominant bits are recorded and then utilized for calculating a constant value to represent the received frame. The values are then accumulated to build the normal model called voltage profile for the transmitter. Viden can then source the transmitter via the generated voltage profile.

The clock skew of electronic devices can also be exploited for fingerprinting the sender. CIDS [13] presents a clock skew-based approach by measuring the arrival time of periodic frames. The deviations between the actual arrival time and the expected time are used for computing the clock skew. However, CIDS can be compromised by adjusting the transmission time to compensate the deviations as demonstrated in [18]. Moreover, the clock skew computed by CIDS is period-dependent [14]. For example, the clock skews of the frames from same ECU with varied length of period might be different computed by CIDS. CANvas [14] fixes the issue by using a greater period called hyper period. Even so, due to the periodical-frame-based nature, they can not be well applied for aperiodic or sporadic frames. Besides, as observed in traffic from real vehicles, some periodic frames have a large deviation from their period or just stop transmission from time to time [14], which can result in performance degradation.

IV. Problem Formulation

The detail meanings and functions of CAN frames are kept confidential in the automotive industry. Specifically, the mapping between CAN frames and sender is usually proprietary and private to which can not be accessed publicly. It becomes a major challenge which hinders the security research on CAN bus [14]. In addition, lacking of sender authentication makes

CAN vulnerable to falsified frames. The adversary can spread the attack easily through the CAN bus by transmitting falsified frames. The problems to be solved in our paper are concluded as follows.

1) Sender Identification: Since there is no any information about the transmitter provided in CAN frames, it is non-trivial to identify the sending ECUs for received CAN frames. The problem can be described as, for a given identifier I_i and a set of corresponding data, our system outputs which ECU E_i the frame of I_i comes from.

2) ECU Enumeration: The ECU enumeration can be applied to detect the changes of network structure, e.g., that an unauthorized node is added or any active nodes are offline from the bus suddenly.

The problem can be formalized as, for a given data traffic, our system outputs the total number of ECUs and the mapping between all extracted identifiers $\{I_1, I_2, \cdots I_m\}$ and thier sending ECUs $\{E_1, E_2, \cdots E_n\}$. Under normal circumstances without attack, the relationship between identifiers and ECU shall be many-to-one [15].

3) Intrusion Detection: The adversary model considered in our work is called masquerade attack [13], [19] launched by injecting malicious CAN frames with forged identifiers. To perform the attack more covertly, the attacker can modify the timing of the transmitting frame to imitate the behaviours of the target ECU. Figure 1 shows how the masquerade attack is mounted. Firstly, the victim ECU (ECU-1) is suspended by stopping transmission, then the attacker (ECU-2) publishes the malicious frames to the bus with forged identifiers at adjusted rates.

Miller and Valasek [20] had proved the feasibility of masquerade attack to control safety-critical functions on real vehicles. To perform the attack on real vehicles, the first priority is to obtain access to the automotive underlying CAN bus. It can be achieved by exploiting vulnerabilities of hardware and software remotely or physically [21]. The attacker can then inject frames with specified semantics (can be revealed by reverse engineering) or fuzz with counterfeit identifiers to compromise another ECU on the bus. Next, the masquerade attack can be mounted to control the vehicle's safety-critical functions.

The reasons why the masquerade attack can be effective are mainly as follows: As the broadcast nature and message-oriented characteristic of CAN protocol (Refer to Section II for details), the decision on how to process the newly received CAN frame only depends on the extracted identifier. Under normal circumstances, the identifier should be unique and can only be sent by one and only one ECU (i.e. the legitimate sender). Thus, the adversary can deceive the receiving ECU by injecting malicious messages with a counterfeit identifier. It is hard to defend against the masquerade attack on CAN since there is no authentication mechanism to verify the sender.

The problem of intrusion detection can be described as, for a given identifier I_i, our system monitors the patterns of traffic of frame I_i continuously. If there is any abnormal deviations (inconsistent sender) found from the normal model of frame I_i, an intrusion is reported.

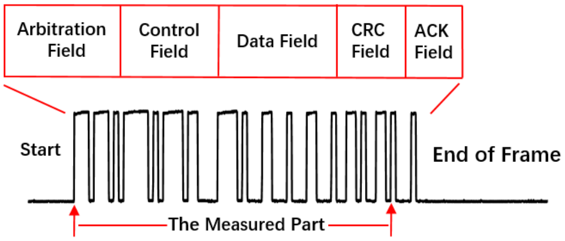


Fig. 2. CAN data frame.

V. Our Method

To solve the above problems, we exploit the difference in signal length to fingerprint the sending ECU. The signal length of the transmitted frame can inherit the deviations caused by the hardware characteristics such as skew in clock and disturbance in transceiver. Thus, we measure the length of the signal for a single CAN frame to extract the deviations. The deviations are then be accumulated to build the model for representing the sender. Since our method works based on characteristics in signal, it is required to have the ability to access the transmission medium (i.e. the wires of CAN bus) of the monitored network directly. If our system fails to access the original electrical signal transmitted on the target CAN bus, we cannot get the leaked information driven by the hardware of its transmitter. From the perspective of our system, all frames from another CAN bus are considered as sent from the gateway node which forwards them. There are five steps in our method as described in the following.

A. Data Collection

In this step, we firstly measure the time duration (denoted by S_i) of the frame-level signal. In the meantime, the identifier I_i and how many bits (denoted by n_i) during the measurement are read by our system as well. The offsets in measured duration from its nominal length (denoted by O_i) is then calculated as: $O_i = S_i - NBT * n_i$, where the NBT refers to the nominal bit time. The product of NBT and the bits number is the nominal length. Next, the data is grouped by its identifier and fed into the next phase.

The measuring process is described as follows. Instead of taking the complete signal into consideration, the measuring is applied to the part of signal only driven by its transmitter. The electrical representation during ACK field does not reflect the characteristics of the transmitter ECU since the transmitter keeps silence during the ACK field. Thus, the signal during the ACK field is excluded from the measuring process. Besides, we measure the time elapsed between two rising edges to simplify the measuring process among different transceivers. In summary, the measurement interval starts from the first rising edge to the second to last rising edge (Figure 2). In our evaluation, we set the starting and ending point of measurement both as $0.9V$ of the rising edge.

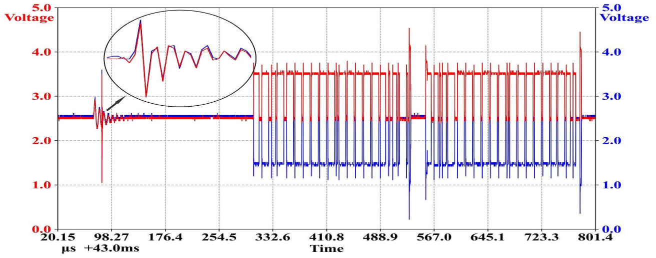


Fig. 3. The differential signal can resist the electromagnetic interference.

The measurement is applied on the the differential signal instead of signal from either CAN high or CAN low. As seen in Figure 3, large fluctuations on each wire can be observed due to the electromagnetic interference. Since the effect of electromagnetic noise on both CAN high and CAN low are nearly identical, the differential signal V can be more immune to such noise and provide a stable electrical signal.

B. Preprocessing

We aim to remove the abnormal measurements (called *outlier*) in this step. The outliers refer to those measurements that deviate too much from the nominal length of the signal. From our observations, there are some measurements collected from real vehicles deviate a lot from the range of normal values. They are introduced by external interference other than the hardware of the transmitter itself. Thus, we propose an novel box-plot algorithm called score-based-boundary-box-plot (SBB-box-plot) to determine the normal range of our data and remove the outliers. The new box-plot algorithm exploits a score-based mechanism to determine the bounds of the box.

The original box-plot [22] is a type of chart which can visually describe the distribution of numerical data and discovery abnormal observations. There are four numbers defined to summarize a set of data: The lower quartile (Q1) is defined as the number below which there are 25% of all numbers. The upper quartile (Q3) means that there are 75% of all numbers lie below this point. The lower whisker can be computed as $Q1 - 1.5 * (Q3 - Q1)$, while the upper whisker is defined as $Q3 + 1.5 * (Q3 - Q1)$. The data points which locate outside the upper and lower whiskers are considered as outliers.

However, the original box-plot algorithm cannot be applied to our problem directly. Figure 4 shows the distribution of the signal length offsets of three CAN frames. The data in Figure 4(a) and 4(b) is collected from the real vehicles, while the Figure 4(c) is from an CAN bus prototype, which is a four-node network established by development boards (Refer to Section VI-A for details). The data is displayed in the form of a histogram. The horizontal axis represents the offsets O_i deviate from the nominal length, while the vertical axis shows the number of frame instances. The data of vertical axis is shown in log scale to emphasize the distribution of offsets with small amount of data. In Figure 4(a) and 4(b), the range of measured offsets is extended very long. Most of the offsets are concentrated around a few consecutive values on the left. These values are regarded as the true offsets affected only by the hardware of the transmitter. Despite the number of data falls into each category on the right is significantly less, the sum of these data is not trivial. This makes the The original box-plot technique inapplicable to our problem. For example, the upper quartile (Q3) of the dataset in Figure 4(a) is calculated as 86 and the upper whisker is 299, which means only a few outliers can be identified. An interesting observation is that outliers can be hardly found in the data from our prototype (Figure 4(c)). It can be explained that the surroundings and the communication traffic of CAN bus prototype is much simpler compared with the CAN bus under the practical circumstances in real vehicles.

Our proposed SBB-box-plot algorithm is presented in Algorithm 1. The main difference lies on how we set the upper and lower boundary of box. We firstly count the frequency over the value of offsets to construct a histogram (Line 1). From our observations, the values which can be observed more frequently are more likely to be true offsets affected only by the transmitter. Thus, the scoring mechanism based on the histogram [23] is adopted in our work. The score is computed based on the inverse of the probability (Line 3 to 4), which means that the lower the score is, the more likely the value of offset is normal. All offsets with scores less than the threshold α are picked and the maximum value is set as the upper boundary (Line 10). In our work, we set $\alpha = 1$. The lower boundary is set to the 10th percentile (Line 11). The reason is as follows. Since the data are measurements from the physical signal, any external interference which may lead to change on the duration or shape of the signal will only lengthen the signal (The signal length would be determined at least by the characteristics of the ECU's own hardware). The data distribution shown in the Figure 4 also corroborates that. Hence, the small measurements matters for determining the boundary of normal data. We set the 10th percentile as the lower boundary. The lower whisker is the lower boundary minus ω times the range of box (Line 14).

C. Clock Skew Estimation

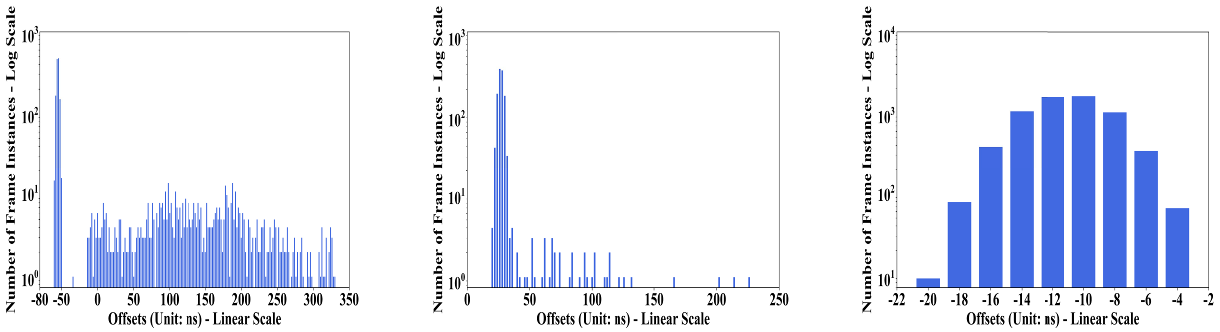
We compute the clock skew in units of frame instances. According to the definition, the clock skew is the difference in frequency (first derivative of its offset) of the clock in ECU with respect to true clock. Considering the clock skew is constant, the accumulated clock offsets within the length of one frame instance increase linearly by time. The slope of the accumulated clock offsets can be regarded as the clock skew. Thus, the clock skew can be computed as offsets (total deviations within the length of one frame instance) divided by the total measured length of this received frame instance. It can be expressed as:

$$Skew_i = \frac{O_i}{S_i} = \frac{S_i - NBT * n_i}{S_i}$$

where, O_i, S_i, NBT, and n_i refer to the signal length offset, the measured length, *Nominal Bit Time* and the corresponding number of bits for the i_{th} frame instance respectively.

D. Model Generation

The process of quantization of measurement can be affected by noise from transmitter or sampling device. It can be referred

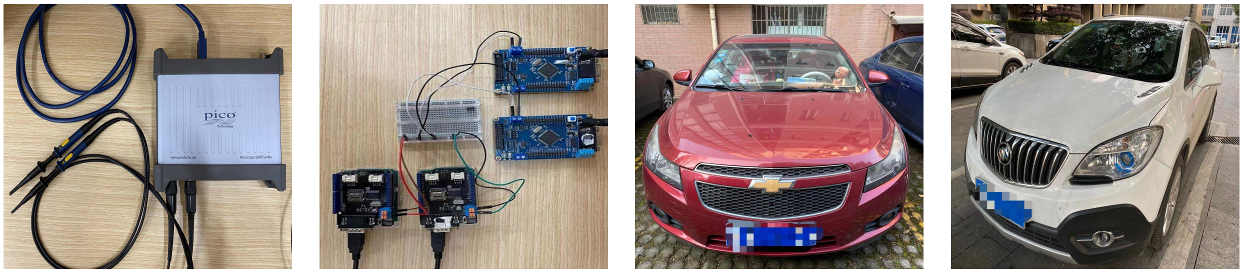


(a) Frame A from real vehicle X.

(b) Frame B from real vehicle Y.

(c) Frame C from prototype.

Fig. 4. The histogram of computed offsets.



(a) Data Collector.

(b) CAN bus prototype.

(c) 2011 Chevrolet Cruze.

(d) 2014 Buick Encore.

Fig. 5. Evaluation platforms, containing (a) PicoScope 5244D for data collection; (b) CAN bus prototype; (c) 2011 Chevrolet Cruze for evaluation; and (d) 2014 Buick Encore for evaluation.

as zero-mean Gaussian noise. To detect small changes and remove the Gaussian noise, the estimated clock skews of same frame are accumulated over the number of frame instances. Since the process is independent with the arrival time of CAN frames, our study can not only be applied to periodic messages but also aperiodic or sporadic messages. We exploit linear regression model to represent each ECU considering the skew in clock is constant. The problem can be formulated as:

$$Skew_{acc}[k] = S[k] \cdot num[k] + e[k],$$

where at step k, $Skew_{acc}[k]$ is the accumulation of clock skews for the first $num[k]$ frame instances, $S[k]$ the regression parameter, $num[k]$ denotes the number of frame instances, and $e[k]$ the identification error. The identification error, e, means the residual which is the difference between the fitted value and the real value (the accumulation of measured clock skews). The slope of the linear model can thus be represented as the clock skew of the transmitter. The Recursive Least Squares (RLS) algorithm [24] is used to fit the data and solve the problem.

E. Application

1) Sender Identification: Given a identifier I_l, the traffic with I_l is measured and recorded to compute the clock skew $Skew_l$ for its transmitter. Then, the estimated clock skew $Skew_l$ is compared with the data in the database of ECUs' clock skew. The matched one is regarded as its sender. An unknown ECU is reported if no one matched.

2) ECU Enumeration: We extract a set of identifiers denoted by $I_{1:m}$ and corresponding estimated clock skews $Skew_{1:m}$ for the monitored network. The identifiers with same clock skews are grouped together as from same ECU. Thus, each group represents an ECU. The total number of groups is the number of ECUs.

3) Intrusion Detection: We detect the intrusion via monitoring two factors, which are cumulative deviations from the linear model and the dropping rates of outliers respectively. Specifically, we firstly collect a set of measurements (regarded as a data block). During the stage of preprocessing, we calculate the dropping rates which is defined as the number of outliers divided by the total number of the data block. If there is a significant increase observed in the dropping rates (greater then the threshold), an anomaly is reported. During the stage of model generation, if any abnormal deviations from the linear model is observed, our system also trigger an alarm. The latter method is necessary. When the relative clock skew between two clocks is small, the data from E_1 can also pass through the box generated by E_2 (the dropping rate would be small).

The identification error-based Cumulative Sum (CUSUM) [13] is adopted to detect the deviations from the linear model. The CUSUM method can detect the change point of a data sequence by accumulating the deviations observed from the target means. The minor deviation can be detected over time due to the feature of cumulative sum. The identification error-based CUSUM is to derive the accumulated deviations from observed identification error e of RLS to the mean identification error $\mu(e)$. Once the

Algorithm 1 Score-Based Boundary Box Plot

```
Input: original data: $\left\{I_k, Offset_k^{original}\right\}$,
$Offset_k^{original}$=[$O_1, O_2, O_3, \cdots, O_n$];
Output: processed data without outliers:
        $\left\{I_k, Offset_k^{processed}\right\}$,
$Offset_k^{processed}$=[$O_1, O_2, O_3, \cdots, O_m$],
$m \leq n$;
Histogram for $Offset_k^{original}$: Count the frequency over
the value of offsets;
while iterate all bins do
    Density Estimation for $i_{th}$ Bin : $p_i$;
    Calculate the Score for $i_{th}$ Bin : $Score_i = \log \frac{1}{p_i}$;
    Find the bins with Score less than threshold $\alpha$:
    if $Score_i \leq \alpha$ then
        Selected_bins.add($Bin_i$);
    end
end
Compute the upper boundary:
$UB = max(Selected\_bins.value\_of\_offset)$;
Compute the lower boundary: $LB = 10_{th}$ Percentile of
$Offset_k^{original}$;
Compute the inter range of Box: $IBR = UB - LB$;
Compute the upper whisker: $UW = UB + w * IBR$;
Compute the lower whisker: $LW = LB - w * IBR$;
Remoue the outliers:
for all data in $Offset_k^{original}$ do
    if $LW \leq O_i \leq UW$ then
        $Offset_k^{processed}$.add($O_i$);
    end
end
```

cumulative sum of deviations exceeds either the upper or lower control limits g^+ and g^-, the intrusion can be detected. The update process of g^+ and g^- is as follows:

$$g^+ \leftarrow max\left[0, g^+ + (e - \mu(e))/\sigma(e) - \beta\right] \quad (1)$$

$$g^- \leftarrow max\left[0, g^- - (e - \mu(e))/\sigma(e) - \beta\right] \quad (2)$$

where $\sigma(e)$ is the standard deviation of identification error e, β is the noise of observations, i.e. the minimum detectable errors. The value of $\mu(e)$ and $\sigma(e)$ can be learned in advance under trusted circumstances.

VI. Evaluation

A. Experimental Settings

1) Data Collector: The PicoScope 5244D (Figure 5(a)) is adopted as our data collector considering its portability and the programmable interfaces provided. Our data collector works at 500 MS/s for sampling speed and 8 bit for sampling resolution.

2) Prototype: We build four-node CAN bus as shown in Figure 5(b). Two nodes employ the Arduino UNO as the motherboard plugged with an CAN module. The transceiver chip is MCP2551. The other two are single development boards integrated with CAN functions. The micro-controller is STM32F103VET6, and the chip of CAN transceiver is TJA1050. The prototype is working at 500 kbps.

3) Real Vehicles: Two production vehicles, including a 2011 Chevrolet Cruze (Figure 5(c)) and a 2014 Buick Encore (Figure 5(d)) are also used for experimental evaluation. We connect our data collector to the internal high-speed (500 kbps) CAN network via OBD-II port. The OBD-II port is a standardized communication port which allows workers to get real-time data from the inside of vehicles and identify the malfunctions.

B. Evaluation on Sender Identification and ECU Enumeration

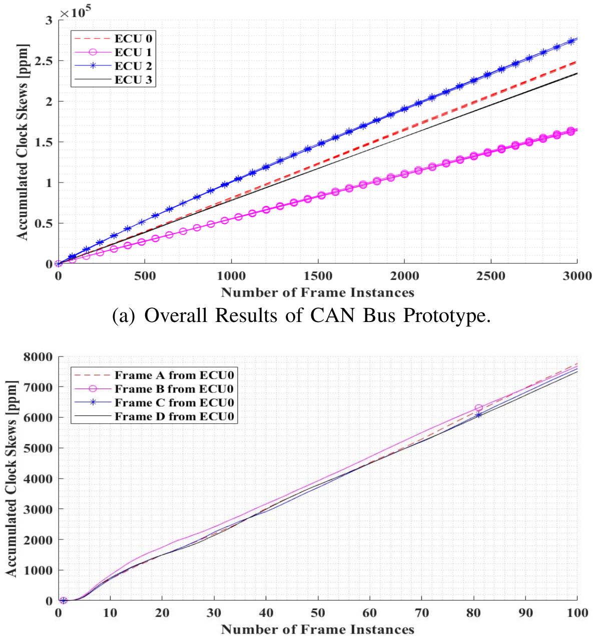
The ECU enumeration on a given CAN bus can be regarded as a problem of identifying the sender for all detected identifiers. Thus, we evaluate the ability of our approach on sender identification and ECU enumeration together in this section. The source mapping built by our approach is then compared with the known ground truth of the corresponding testbeds. It should be noted that the number of recorded instances for each frame can be different since the period or frequency of frames is different. In this section, the process of data preprocessing is based on the whole recorded data. The forgetting factor λ of RLS is set as 0.9995.

1) Prototype: We set that each ECU on the prototype sends four different frames. The four frames from the same node are transmitted in a random order. Besides, the data length of CAN frames (i.e. the number of bytes of data) and the payload are also randomly generated.

The results can be seen in Figure 6. Figure 6(a) shows the accumulated clock skews for all frames. We can see that, the four ECUs are clearly identified. The lines of the same shape and color (represent the four frames from the same ECU) are almost overlapped. Furthermore, the results of first 100 frame instances of four frames sent by ECU0 are drawn in the Figure 6(b). The results on the prototype show that our approach can well identify whether multiple frames are sent from the same ECU, as well as distinguish the frames coming from different ECUs.

It should be noted that there is no any outliers found in the data recorded from the CAN bus prototype. This observation supports our statement that the abnormal measurements of time offsets are mainly contributed by the interference from the heavy traffic and complicated surroundings of the vehicles.

2) Chevrolet Cruze: Figure 7 shows the results on Chevrolet Cruze. To emphasize the importance of data preprocessing, the results without preprocessing are also presented as shown in the left sub-figure, i.e. Figure 7(a). There are four ECUs in total denoted as ECU0 to ECU3 on the CAN network (connected to the OBD-II port directly) of Cruze. From the Figure 7(a), the accumulation of computed clock skews based on the raw data can hardly reflect the characteristics of clock skew of the transmitter. Data with different identifiers from the same sending node do NOT exhibit the same clock skew, which means that it can hardly be used for pinpointing the sender. Instead, those lines which represent identifiers from



(a) Overall Results of CAN Bus Prototype.

(b) Results of Frames Sent by ECU0.

Fig. 6. Accumulated clock skews over the number of frame instances of CAN bus prototype.

same ECU almost coincide with each others in the Figure 7(b). Besides, the frames came from different ECUs can be well distinguished. It can be easily found that there are four ECUs detected by our approach. The comparison of results with/without data preprocessing shows that the SBB-box-plot algorithm can be effective for removing the abnormal measurements observed under the practical conditions (real vehicle).

The relationship between the ECU and the number of unique identifiers (i.e. how many lines in the figure) is presented in Table I. There are 52 frames (i.e. 52 unique IDs) detected in total, and 17 frames at most sent by one ECU (ECU2). Considering the data length (i.e. the number of bytes of data frame) is constant over time [13] which is also confirmed by our observations, how many frames with a given data length and the proportion are presented in the Table III. We can see that, the data length of most (more that 50%) frames is 8 bytes. This can be expected considering the size of payload of CAN frame is very limited and to maximize utilization is reasonable. There are also more than 20% of frames with payload less than 4 bytes. Despite there are up to 52 frames with data length varied from 1 to 8 bytes transmitted on the bus, our approach can work effectively.

TABLE I

THE ECU AND ITS NUMBER OF UNIQUE IDs ON **CHEVROLET CRUZE**

ECU#	Number of ID
ECU0	15
ECU1	14
ECU2	17
ECU3	6
Total	52

TABLE II

THE ECU AND ITS NUMBER OF UNIQUE IDs ON **BUICK ENCORE**

ECU#	Number of ID
ECU A	20
ECU B	22
ECU C	28
ECU D	9
Total	79

TABLE III

THE NUMBER OF IDs WITH A GIVEN DATA LENGTH AND ITS SHARE - **CHEVROLET CRUZE**

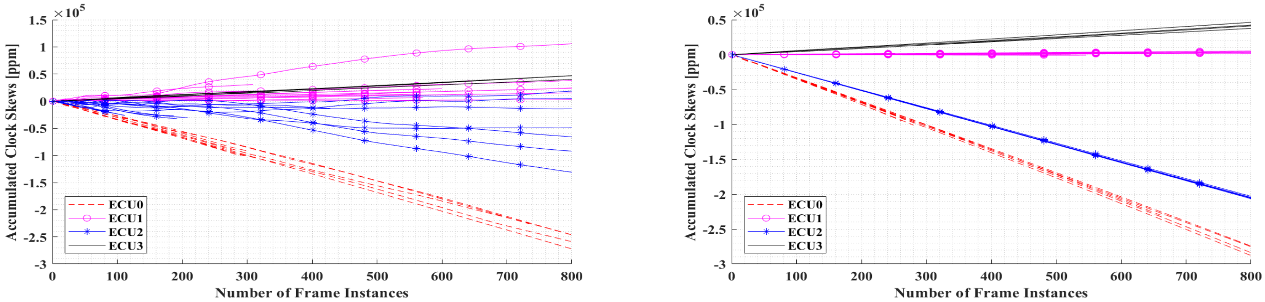
Data Length	Number of ID	Shares (%)
1	1	1.92
2	0	0
3	4	7.69
4	8	15.38
5	1	1.92
6	4	7.69
7	5	9.62
8	29	55.77

TABLE IV

THE NUMBER OF IDs WITH A GIVEN DATA LENGTH AND ITS SHARE - **BUICK ENCORE**

Data Length	Number of ID	Shares (%)
1	2	2.53
2	5	6.33
3	4	5.06
4	7	8.86
5	6	7.59
6	9	11.39
7	10	12.66
8	36	45.57

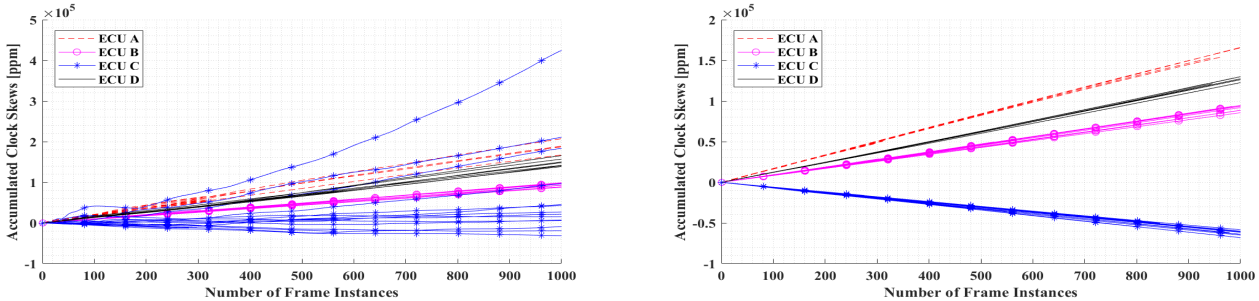
3) Buick Encore: The Figure 8(a) shows the fitted lines of accumulated clock skews based on the raw measurements. Similar to the results of Chevrolet Cruze, the sender can NOT be sourced, as well as the ECUs are indistinguishable. After removing the outliers, the normal model of clock skews can be constructed clearly. The relationship between the ECU and the number of frames is shown in Table II. And the number of frames with a given data length and its shares on all frames is described in Table IV. There are totally 79 different frames detected which is larger than on Chevrolet Cruze. Similarly, the data length of frames varies from 1 to 8 bytes. The frames sent from same ECU can exhibit nearly same clock skews. The difference among different ECUs can also be easily recognized. The results on Buick Encore shows again that our approach can works well on source identification and ECU enumeration without any prior information of the target network. Not only on the prototype, our approach can also work well on automotive network under complicated working conditions.



(a) Results of Chevrolet Cruze without Data Preprocessing.

(b) Results of Chevrolet Cruze with Data Preprocessing.

Fig. 7. Accumulated clock skews over the number of frame instances of Chevrolet Cruze.



(a) Results of Buick Encore without Data Preprocessing.

(b) Results of Buick Encore with Data Preprocessing.

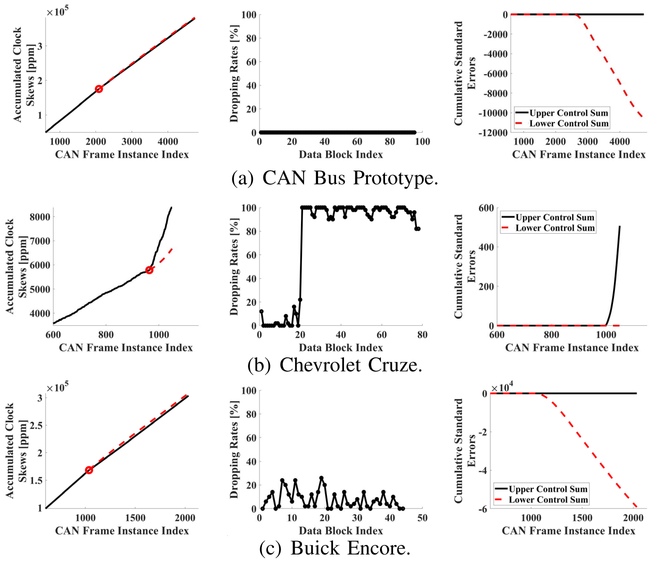
Fig. 8. Accumulated clock skews over the number of frame instances of Buick Encore.

C. Evaluation on Intrusion Detection

The ability of our approach against the masquerade attack is evaluated in this section. We simulate the scenarios when masquerade attack occurs by data collecting from the evaluation platforms. Firstly, two ECUs of each evaluation platform are selected as the victim ECU and the attacker respectively. Two data segments (denoted by $Data_{victim}$ and $Data_{attacker}$) with given identifiers are collected from the victim ECU and the attacker respectively. Then, the $Data_{victim}$ is divided into two parts. The first part $Data_{victim}^{training}$ is used for training to generate the shape of the box based on the SBB-Box-plot algorithm, and learn the value of $\mu(e)$ and $\sigma(e)$ for CUSUM detection. Next, the second part (denoted by $Data_{victim}^{testing}$) of the data segment from the victim ECU is concatenated with $Data_{attacker}$ (data segment from the attacker) to form a new data segment $Data^{testing}$ for testing. A flag is added for each single data point in $Data^{testing}$ to indicate whether it is from the attacker ECU as the ground truth. The junction between the two original data segments can be regarded as when the masquerade attack is mounted. In our evaluation, the size N for each data block is set as $N = 50$. The threshold for monitoring the dropping rates is set as 70%. The noise β and threshold of CUSUM method is set both as 20 standard errors. The forgetting factor λ of RLS is set as 0.9995.

1) Prototype: For prototype, we set the ECU0 as the victim and ECU3 as the attacker considering the relative clock skews between them is the smallest among the four ECUs according to the results in Section VI-B.1. The length of training data is set as 2000 frame instances. The evaluation results of intrusion detection is shown in the Figure 9(a). The left figure shows the accumulated clock skews of the concatenated data segment $Data^{testing}$. The red point marked in the figure indicates when the masquerade attack is mounted. The second picture of Figure 9(a) is the dropping rates for every newly received data block. We can see that, no matter the normal data or the malicious data, the dropping rates remains at 0%. The reason is that none of the outliers can be detected from our prototype as discussed in the Section VI-B.1. For the malicious data, the generated box is wide enough for passing all data from both ECUs considering the relative clock skew between these two ECUs is small. Despite the detection by monitoring the dropping rates fails, the cumulative sum of deviations from the expected accumulated clock skews can be effective for intrusion detection. The right figure presents the results of CUSUM method. The cumulative sum of deviations exceeds the lower control limit after the attack occurs. The results demonstrate that our method can defend against the masquerade attack on the prototype.

2) Real Vehicles: The evaluation results on real vehicles are shown in the Figure 9(b) and 9(c). The ECU1 and ECU0 are regarded as the victim and attack ECU for Chevrolet Cruze, and ECU A and ECU D are victim and attacker for Buick Encore. One of the frames from each ECU is logged for the evaluation. The size of the training data is set as 800 frame instances on both vehicles. For Chevrolet Cruze, the intrusion can be detected effectively by monitoring the dropping rates. The middle picture in the Figure 9(b) shows that the box generated by the victim ECU can find (filter out) most of



(a) CAN Bus Prototype.

(b) Chevrolet Cruze.

(c) Buick Encore.

Fig. 9. Evaluation results for intrusion detection, containing: (1) The left figures for accumulated clock skews of victim frame. The red point is to mark when the masquerade attack is mounted. (2) The middle figure indicates the dropping rates of data blocks. (3) The right figure is the CUSUM chart of identification error e.

the data from the attacker. The size of the data block is set as 50 in our evaluation, which means that the alarm can be triggered within 50 frame instances. Similarly, the Figure 9(c) shows one example for intrusion detection on Buick Encore. Despite the dropping rates do not exceed the threshold after attack, the identification error-based CUSUM can well take effect. As shown in the right and left sub-figures, the attack is mounted at the point of 1037_{th} frame instance, and the alarm is triggered at the point of 1051_{th} frame instances which means the gap is 14 filtered messages since the attack is mounted. The evaluation results on both the prototype and real vehicles show that our approach can be effective for detecting masquerade attack. Since the process of detection is independent from the arrival time or transmission time of the received frames, our approach which can be applied on both periodic and aperiodic traffic outperforms existing clock-skew-based methods [13], [14].

VII. Conclusion

Besides the huge convenience and efficiency brought by the IoT, the security becomes one major challenge faced by the ITS. Attack can be spread to the safety-critical automotive components easily due to the intrinsic vulnerabilities in CAN. To protect CAN bus, we propose a model-based approach which utilizes the clock skew to enable identifying the sending ECU and detecting attack. Comparing with those supervised learning algorithms based methods, our decision process can be clear and explainable. Besides, our approach is based on the measurements of the frame-level signal. Thus, it can be applied on periodic frames as well as aperiodic or sporadic frames. To show the feasibility of our approach, the evaluation on one four-node CAN network established by development boards as well as two production vehicles is performed. The results validate that the linear model can well represent the transmitter for received CAN frames and can be differentiated among ECUs on the proprietary CAN without prior knowledge. Also, the intrusion can be triggered effectively. In conclusion, we provide an effective and feasible way capable of improving the security of CAN. This can further enhance the security of the ITS.

References

[1] F. Zhu, Y. Lv, Y. Chen, X. Wang, and F. Wang, “Parallel transportation systems: Toward IoT-enabled smart urban traffic control and management,” *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 10, pp. 4063–4071, Oct. 2020.

[2] Z. Zhou, A. Gaurav, B. B. Gupta, M. D. Lytras, and I. Razzak, “A fine-grained access control and security approach for intelligent vehicular transport in 6G communication system,” *IEEE Trans. Intell. Transp. Syst.*, early access, Sep. 2, 2021, doi: 10.1109/TITS.2021.3106825.

[3] M. M. Hussain and M. M. S. Beg, “Using vehicles as fog infrastructures for transportation cyber-physical systems (T-CPS): Fog computing for vehicular networks,” *Int. J. Softw. Sci. Comput. Intell.*, vol. 11, no. 1, pp. 47–69, Jan. 2019.

[4] M. Serror, S. Hack, M. Henze, M. Schuba, and K. Wehrle, “Challenges and opportunities in securing the industrial Internet of Things,” *IEEE Trans. Ind. Informat.*, vol. 17, no. 5, pp. 2985–2996, May 2021.

[5] K. Koscher *et al.*, “Experimental security analysis of a modern automobile,” in *Proc. IEEE Symp. Secur. Privacy*, May 2010, pp. 447–462.

[6] Y. Chen, W. Hu, M. Alam, and T. Wu, “Fiden: Intelligent fingerprint learning for attacker identification in the industrial Internet of Things,” *IEEE Trans. Ind. Informat.*, vol. 17, no. 2, pp. 882–890, Feb. 2021.

[7] P.-S. Murvay and B. Groza, “Source identification using signal characteristics in controller area networks,” *IEEE Signal Process. Lett.*, vol. 21, no. 4, pp. 395–399, Apr. 2014.

[8] W. Choi, H. J. Jo, S. Woo, J. Y. Chun, J. Park, and D. H. Lee, “Identifying ECUs using inimitable characteristics of signals in controller area networks,” *IEEE Trans. Veh. Technol.*, vol. 67, no. 6, pp. 4757–4770, Jun. 2018.

[9] 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 Trans. Inf. Forensics Security*, vol. 13, no. 8, pp. 2114–2129, Aug. 2018.

[10] M. Kneib and C. Huth, “Scission: Signal characteristic-based sender identification and intrusion detection in automotive networks,” in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, New York, NY, USA, Oct. 2018, pp. 787–800.

[11] M. Kneib, O. Schell, and C. Huth, “EASI: Edge-based sender identification on resource-constrained platforms for automotive networks,” in *Proc. Netw. Distrib. Syst. Secur. Symp.*, San Diego, CA, USA, 2020, pp. 1–16.

[12] J. Zhou, P. Joshi, H. Zeng, and R. Li, “BTMonitor: Bit-time-based intrusion detection and attacker identification in controller area network,” *ACM Trans. Embedded Comput. Syst.*, vol. 18, no. 6, pp. 1–23, Nov. 2019.

[13] K.-T. Cho and K. G. Shin, “Fingerprinting electronic control units for vehicle intrusion detection,” in *Proc. 25th USENIX Conf. Secur. Symp.* Berkeley, CA, USA: USENIX Association, 2016, pp. 911–927.

[14] S. Kulandaivel, T. Goyal, A. K. Agrawal, and V. Sekar, “CANvas: Fast and inexpensive automotive network mapping,” in *Proc. 28th USENIX Conf. Secur. Symp.* Berkeley, CA, USA: USENIX Association, 2019, pp. 389–405.

[15] M. Di Natale, H. Zeng, P. Giusto, and A. Ghosal, *Understanding and Using the Controller Area Network Communication Protocol: Theory and Practice*. New York, NY, USA: Springer, 2012.

[16] *Pll Jitter and its Effects in the CAN Protocol*, TB078, Microchip Technol. Inc., Chandler, AZ, USA, datasheet, 2004.

[17] K.-T. Cho and K. G. Shin, “Viden: Attacker identification on in-vehicle networks,” in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2017, pp. 1109–1123.

[18] X. Ying, S. U. Sagong, A. Clark, L. Bushnell, and R. Poovendran, “Shape of the cloak: Formal analysis of clock skew-based intrusion detection system in controller area networks,” *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 9, pp. 2300–2314, Sep. 2019.

[19] 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 *Proc. 35th Annu. Comput. Secur. Appl. Conf.*, Dec. 2019, pp. 229–244.

[20] C. Miller and C. Valasek, “Remote exploitation of an unaltered passenger vehicle,” *Black Hat USA*, vol. 2015, p. 91, Aug. 2015.

[21] S. Checkoway *et al.*, “Comprehensive experimental analyses of automotive attack surfaces,” in *Proc. 20th USENIX Conf. Secur.*, Berkeley, CA, USA: USENIX Association, 2011, p. 6.

[22] D. F. Williamson, R. A. Parker, and J. S. Kendrick, “The box plot: A simple visual method to interpret data,” *Ann. Internal Med.*, vol. 110, no. 11, pp. 916–921, 1989.

[23] M. Goldstein and A. Dengel, “Histogram-based outlier score (HBOS): A fast unsupervised anomaly detection algorithm,” in *Proc. 35th German Conf. Artif. Intell.*, 2012, pp. 59–63.

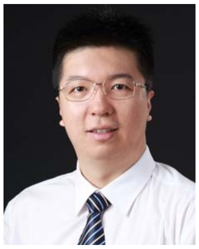
[24] S. S. Haykin, *Adaptive Filter Theory*. London, U.K.: Pearson, 2008.

Jia Zhou received the Ph.D. degree in computer science and technology from Hunan University in 2021. He is currently working as a Post-Doctoral Researcher at the Peng Cheng Laboratory. His research interests include automotive security, cyber-physical systems, and embedded systems.

Guoqi Xie (Senior Member, IEEE) received the Ph.D. degree in computer science and engineering from Hunan University in 2014.

He was a Post-Doctoral Research Fellow with Nagoya University. He is currently a Professor at Hunan University. His current research interests include real-time systems, automotive embedded systems, and embedded safety and security. He is an ACM Senior Member. He received the 2018 IEEE TCSC Early Career Researcher Award. He is also serving on the Editorial Boards for *Journal of Systems Architecture*, *Microprocessors and Microsystems*, and *Journal of Circuits, Systems and Computers*.

Haibo Zeng (Member, IEEE) received the Ph.D. degree in electrical engineering and computer sciences from the University of California at Berkeley. He was a Senior Researcher with General Motors Research and Development until October 2011 and an Assistant Professor with McGill University until August 2014. He is currently with the Department of Electrical and Computer Engineering, Virginia Tech. His research interests include embedded systems, cyber-physical systems, and real-time systems. He received four best paper/best student paper awards in the above fields.



Weizhe Zhang (Senior Member, IEEE) received the B.Eng., M.Eng., and Ph.D. degrees in engineering (computer science and technology) from the Harbin Institute of Technology, China, in 1999, 2001, and 2006, respectively. He is currently a Professor with the School of Computer Science and Technology, Harbin Institute of Technology, and the Director of the Cyberspace Security Research Center, Peng Cheng Laboratory, Shenzhen, China. He has published more than 100 academic papers in journals, books, and conference proceedings. His research interests include parallel computing, distributed computing, cloud and grid computing, and computer network.

Laurence T. Yang (Fellow, IEEE) received the B.E. degree in computer science and technology and the B.S. degree in applied physics from Tsinghua University, Beijing, China, in 1992, and the Ph.D. degree in computer science from the University of Victoria, Victoria, BC, Canada, in 2006. He is currently a Professor with St. Francis Xavier University, Antigonish, NS, Canada. His research has been supported by the National Sciences and Engineering Research Council, and the Canada Foundation for Innovation. His research interests include parallel and distributed computing, embedded and ubiquitous/pervasive computing, and big data.

Mamoun Alazab (Senior Member, IEEE) received the Ph.D. degree in computer science from the School of Science, Information Technology and Engineering, Federation University of Australia. He is currently an Associate Professor with the College of Engineering, IT and Environment, Charles Darwin University, Australia. He is also a Cyber Security Researcher and a Practitioner with industry and academic experience. His research is multidisciplinary that focuses on cyber security and digital forensics of computer systems with a focus on cybercrime detection and prevention, including cyber terrorism and cyber warfare. He works closely with government and industry on many projects, including the Northern Territory (NT) Department of Information and Corporate Services, IBM, Trend Micro, and the Australian Federal Police (AFP). He is the Founder and the Chair of the IEEE Northern Territory (NT) Subsection Detection and Prevention.

Renfa Li (Senior Member, IEEE) is currently a Professor of computer science and electronic engineering with Hunan University, Changsha, China. He is also the Director of the Key Laboratory for Embedded and Network Computing of Hunan Province, China. His current research interests include computer architectures, embedded computing systems, cyber-physical systems, and the Internet of Things. He is a member of the Council of CCF and a Senior Member of ACM.